



Universidad  
Carlos III de Madrid

Departamento de Informática

Ingeniería en Informática

PROYECTO FIN DE CARRERA

# APLICACIÓN MÓVIL BASADA EN REDES SOCIALES PARA DAR SOPORTE A LA REALIZACIÓN DE EXÁMENES TIPO TEST (PLATAFORMA IOS)

Autor: D. Roberto Estrada Casarrubios

Director: Dr. Javier García Guzmán

Tutor: Dr. Alberto Heredia García

Leganés, 31 Octubre de 2013



**Título:** Aplicación móvil basada en redes sociales para dar soporte a la realización de exámenes tipo test (plataforma iOS)

**Autor:** D. Roberto Estrada Casarrubios

**Director:** Dr. Javier García Guzmán

## EL TRIBUNAL

**Presidente:** D. José Luis López Cuadrado

**Vocal:** Dr. Raúl Sánchez Reíllo

**Secretario:** D. Alejandro Ruiz Robles

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 31 de Octubre de 2013 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE





---

## Agradecimientos

Aún recuerdo mi primer día en la Universidad, aquel día llegué a casa pálido como este papel y pensé que aquello no era para mí. Toda mi vida peleando para llegar hasta ahí y ahora quería retirarme. Mis padres me miraron sorprendidos, no entendían que sucedía y al instante supe que no podía permitirme decepcionarles por todo lo que habían hecho siempre para que pudiera cumplir mis metas, por ello decidí seguir y ahora me encuentro aquí, redactando estas líneas para ellos. Gracias por todo el apoyo incondicional que me habéis dado y por vuestro esfuerzo a lo largo de todos estos años.

Quiero dar las gracias a toda mi familia por interesarse siempre por mis progresos y compartir mis alegrías y decepciones. Gracias a todos mis tíos y primos. Gracias también a toda mi familia cántabra primero por todo el aprecio que siempre me han tenido y por todo el apoyo que me han brindado, en especial durante las últimas semanas de la elaboración de este proyecto.

He de agradecer a todos mis compañeros y amigos de clase que me han acompañado a lo largo de todos estos años pero especialmente a los siguientes. A Adrián y Luis por ser excelentes personas y que siempre se han preocupado de echarnos una mano desde Delegación de Estudiantes cuando lo hemos necesitado. A Álvaro y Castre mis primeros amigos de la carrera con los que he vivido infinidad de anécdotas. A Coral, mi fiel e incansable compañera de prácticas, amiga como pocas y probablemente la persona más capaz con la que haya trabajado nunca. A Fran, por fundar y contar conmigo en el equipo de fútbol sala con el que tanto hemos disfrutado. A Laura, la luz y la alegría de mis momentos más oscuros. A Lidia, por toda la bondad que siempre ha tenido hacia mí incluso en las situaciones más complejas y a Vanesa, por tener siempre una sonrisa o un buen consejo que darme. Tampoco quiero dejarme en el tintero a buenas personas como Alberto, Alexia, Chris, Carlota, Dani, Jorge, Sohrab, Sofía o Zaira.

También quiero dar las gracias a mis amigos de siempre, con los que he crecido por los buenos momentos que he vivido con ellos a lo largo de este periodo de mi vida y ojalá que por muchos años más. Adrián, Borja, Cristina, Desiré, Gonzalo, Iván, Juan David, Miguel, Ricardo y Rocío. Gracias amigos.

Gracias también a ti, Lina, por tu altruismo y por ayudarme sin condiciones siempre que lo he necesitado, llegando a pagar en ocasiones un alto precio personal. Gracias por tu incansable apoyo.

No puedo pasar sin agradecer todo este tiempo que he pasado trabajando con la familia de miSEL. Gracias a mi director, Javier García Guzmán y a mi tutor Alberto Heredia García por hacerme un hueco en el laboratorio. Gracias a los chicos, Javier, José y Roberto por la acogida que me dieron y lo fácil que me han hecho la vida y por ser uno más. He aprendido mucho con ellos y estoy seguro que ellos también de mí, ha sido un verdadero honor y un placer trabajar junto a vosotros. Me llevo tres buenos amigos.

No puedo olvidarme tampoco de mis amigos de la ciudad del Turia a los que conocí por un gran cúmulo de casualidades y que aunque desde lejos y viéndoles en contadas ocasiones siempre han ejercido como lo que son, grandes amigos. Gracias Paco, Alma, Noemí y Rebeca y sobre todo, gracias María Dolores, porque siempre has acudido cuando te he necesitado, porque si no nos hubiéramos tropezado yo no sería quien soy. Tú me enseñaste que la vida no está para darle tantas vueltas y que a veces hay que armarse de valor y tirar por la calle de en medio, que la solución a los problemas no consiste en esperar a que venga alguien que los solucione por ti, sino que has de ser tu quien tome la iniciativa. Que haya llegado hasta aquí es —en gran parte— culpa tuya.

Mi reconocimiento también para Pablo y Rubén y a todos los amigos que he conocido este 2013 gracias a al voluntariado en el mundial de balonmano Handball 2013.

También quiero expresar mi gratitud hacia mis tres pequeñas y preciosas sobrinas de las que tan orgulloso me siento. Elisa, Jimena, Claudia, esto también va por vosotras.

Para concluir, quiero dedicar este proyecto fin de carrera a la memoria de Casilda, mi abuela materna, quien desafortunadamente dejó de estar entre nosotros durante las etapas finales de la realización de este proyecto y no pudo ver cumplido mi objetivo. Doy gracias por todas sus enseñanzas y valores que me transmitió desde la niñez, los cuales me han conducido siempre por la buena senda a lo largo de este largo camino de esfuerzo y sacrificio personal.





*"I saw my death in my dreams many times,*

*but I didn't die.*

*I was better than my fears.*

*Better than my nightmares,*

*but to find out, I had to face them all.*

*I had to get through the worst, to prove I was the best.*

*That was my reality."*

Col. William 'Warwolf' Bishop (USAF) — *Ace Combat: Assault Horizon*



---

## Resumen

En un panorama tecnológico marcado por la aparición de las redes sociales y la omnipresencia de dispositivos móviles como *smartphones* y tabletas, el mundo educativo se enfrenta al desafío de la incorporación de estos nuevos recursos al proceso docente.

Este proyecto surge como respuesta a la necesidad de los docentes de emplear estos nuevos recursos tecnológicos para la enseñanza. La mera posibilidad de interactuar con el alumnado, sugiriendo contenidos, resolviendo dudas, proponiendo debates en cualquier momento y lugar era algo impensable hasta hace algunos años, pero ahora, gracias al avance tecnológico es perfectamente posible.

El objetivo principal de este proyecto consiste en proveer una herramienta a profesores que les permita poder preparar por adelantado a una clase todos los recursos interactivos que se vayan a necesitar durante dicha clase para que el profesor pueda dedicarse a lo que mejor sabe hacer, enseñar. También se busca crear una herramienta que sea de utilidad para los alumnos, sobre todo a la hora de participar en clase y mejorar su capacidad de atención, pero también a la hora de proporcionar un valioso feedback a quien dirige la clase y que le permita tomar decisiones acerca de qué rumbo debe tomar la sesión pudiendo hacer hincapié en aquellos conceptos que no se han comprendido adecuadamente.

Los profesores a través de esta herramienta pueden hacer uso de las redes sociales preparando el contenido que se publicará en ellas durante las clases. Por ejemplo, si un profesor desea publicar tweets con consejos útiles, conceptos básicos o aquella información que desee puede prepararlos con antelación y, una vez en la clase y llegado el momento publicar la información con un simple toque en su dispositivo. Lo mismo sucede con las preguntas de test que quiera llevar a cabo, se preparan previamente y llegado el momento se envían a los alumnos quienes contestarán a través de sus dispositivos.

Los alumnos a través de sus dispositivos podrán seguir ese flujo de información adicional que se publique en las redes sociales, contribuir a él ampliándolo o proponiendo un debate que una vez finalizada la clase todo el mundo podrá volver a leer y extraer la información que más útil resulte.

Los alumnos también a través de sus dispositivos podrán responder a las preguntas de tipo test que el profesor les formule. Para identificar las respuestas y realizar evaluaciones u obtener estadísticas se empleará el perfil de la red social Twitter.

Este proyecto busca también sentar las bases de todo un sistema de aprendizaje personal basado en dispositivos móviles y redes sociales, y que está siendo investigado y desarrollado por el grupo de *mobile intelligence* (miSEL) perteneciente al *Software Engineering Lab* de la Universidad Carlos III de Madrid.

Dicho sistema pretende proporcionar una serie de herramientas para facilitar el aprendizaje y la enseñanza a través de la cooperación entre alumnos y profesores

**Palabras clave:** enseñanza, planificación de clases, redes sociales, Twitter, dispositivos móviles



---

## Abstract

In a technological panorama marked by the social networks expansion and the omnipresence of smartphones and tablets, the academic world faces the challenge of incorporating these new resources to the educational process.

This project arises as response to the teacher's need to use these new technical resources for teaching. The mere possibility of interacting with the students, suggesting contents, clearing doubts or opening debates anytime, anywhere is something that was almost unthinkable until recent years, but now, thanks to the advance of tech is perfectly possible.

The main objective of this project consists in providing a tool to the teachers to allow them to prepare in advance of a class all the necessary resources that they are going to need during such class so that the teacher can dedicate to the best he knows: teaching. The creation of a tool that is useful for the students is also wanted, mostly at the time of taking part in the class and to improve their attention, but also at the time of providing valuable feedback to the one who runs the class so that it allows him to decide which course the session should take, being able to emphasize those concepts that are not very clear.

Teachers can take advantage from social media through this tool by preparing the content that will be posted during classes, i.e. if a teacher wants to post tweets with tips, basics or whatever information he wants, he can prepare them in advance and, once at the class and at the proper time, post the information with a simple touch on his device. Same applies to the quizzes that the teacher may want to run, they are prepared in advance and sent during the class to the students whom will answer through their devices.

The students will be able to keep an eye through their devices to that new additional information stream at the social media, contribute to it or opening a debate that once the class has finished everybody can check it out and revisit it to extract useful information. Students will also be able to participate at the quizzes that the teacher may run. To identify the answers, evaluate them or getting statistics, the Twitter profile of the student will be employed.

This project wants to settle the basis of a whole, brand new mobile and social media based personal learning environment, which is being researched and developed by the *mobile intelligence* group (miSEL), part of the *Software Engineering Lab* from Universidad Carlos III de Madrid. This system seeks to provide a series of tools to ease learning and teaching through cooperation between students and teachers.

**Keywords:** teaching, class planning, social media, Twitter, mobile devices.



## Tabla de contenidos

<b>Capítulo 1: Introducción .....</b>	<b>19</b>
1.1 Contexto .....	20
1.2 Problema .....	21
1.2.1 Profesor.....	21
1.2.2 Alumno .....	21
1.2.3 Claves del problema .....	21
1.3 Motivación del proyecto.....	22
1.4 Objetivos del proyecto.....	22
1.5 Método de resolución .....	23
1.6 Definiciones, acrónimos y abreviaturas .....	24
1.6.1 Definiciones.....	24
1.6.2 Acrónimos .....	24
1.6.3 Abreviaturas .....	24
1.7 Visión general del documento .....	25
<b>Capítulo 2: Estado del arte .....</b>	<b>27</b>
2.1 Introducción a los entornos de aprendizaje personal.....	28
2.2 Análisis de aplicaciones relacionadas con los objetivos establecidos .....	29
2.2.1 EduClick .....	29
2.2.2 Student Response System – SRS.....	29
2.2.3 Blackboard .....	30
2.2.4 Moodle .....	32
2.2.5 Comparativa entre aplicaciones .....	32
2.3 Elección de la plataforma de desarrollo .....	33
2.4 El entorno de desarrollo iOS .....	34
2.4.1 Estructura de iOS.....	34
2.4.2 El lenguaje de programación Objective-C .....	35
2.4.3 El entorno de desarrollo integrado XCode.....	35
2.4.4 La gestión de dependencias y CocoaPods: supervivencia en un mar de caos.....	36
2.5 Desarrollo ágil de software .....	37
2.5.1 Scrum.....	37
2.5.2 Kanban .....	38
2.5.3 Scrumban.....	39
2.5.4 miSEL Software Development Process (miSEL-sdp) .....	40
<b>Capítulo 3: Análisis de la solución .....</b>	<b>41</b>
3.1 Introducción .....	42
3.2 Descripción general del sistema a desarrollar .....	42
3.2.1 Perspectiva del producto.....	42
3.2.2 Funciones del producto .....	42
3.2.3 Características de los usuarios.....	43
3.2.4 Restricciones.....	43
3.2.5 Suposiciones y dependencias .....	43
3.3 Alcance de la solución.....	44
3.4 Historias de usuario.....	44
3.4.1 Product backlog.....	45
3.5 Especificación de requisitos .....	46
3.5.1 Formato de los requisitos .....	46
3.5.2 Catálogo de requisitos .....	46
3.5.3 Trazabilidad de requisitos – historias de usuario.....	53
3.6 Especificación de casos de uso .....	54
3.6.1 Diagrama de casos de uso .....	54
3.6.2 Descripción textual .....	55
3.6.3 Trazabilidad requisitos – casos de uso.....	58

<b>Capítulo 4: Diseño de la solución</b>	<b>59</b>
4.1 Diseño arquitectónico del sistema	60
4.1.1 Primera alternativa	60
4.1.2 Segunda alternativa	60
4.1.3 Tercera alternativa	61
4.1.4 Diagrama de componentes	63
4.2 Diseño detallado del sistema	64
4.2.1 Aplicación de preparación de clases	64
4.2.2 Aplicación del alumno	70
4.2.3 Servicios Web	73
4.3 Servicios Web: Especificación de la API REST del servicio de cuestionarios	74
4.3.1 Sesión	74
4.3.2 Alumnos	77
4.3.3 Preguntas	80
4.4 Diagramas de secuencia	83
4.4.1 Gestión de cursos	83
4.4.2 Gestión de sesiones	85
4.4.3 Gestión de preguntas	87
4.4.4 Gestión de conceptos	89
4.4.5 Gestión de tweets	91
4.4.6 Consulta de estadísticas	93
4.4.7 Suscribirse a una sesión	94
<b>Capítulo 5: Pruebas del sistema</b>	<b>95</b>
5.1 Especificación de pruebas del sistema	96
5.1.1 Formato de las pruebas	96
5.1.2 Catálogo de pruebas	96
5.1.3 Trazabilidad de pruebas de sistema – requisitos funcionales	103
<b>Capítulo 6: Presupuesto</b>	<b>105</b>
6.1 Presupuesto del proyecto	106
6.1.1 Costes de personal	106
6.1.2 Costes de materiales	106
6.1.3 Otros costes	107
6.1.4 Coste total	107
<b>Capítulo 7: Conclusiones y líneas futuras</b>	<b>109</b>
7.1 Conclusiones	110
7.2 Líneas futuras	111
7.2.1 Adaptación a iOS 7	111
7.2.2 Subida de ficheros, videos y otros recursos	112
7.2.3 Mejora del sistema de preguntas	112
7.2.4 Integración con otros sistemas LMS	112
7.2.5 Integración con otras redes sociales	112
7.2.6 Mejoras de <i>backend</i>	112
<b>Referencias</b>	<b>113</b>
<b>Apéndice A: Manual de usuario</b>	<b>115</b>
A.1 mTeachPlan: Manual del profesor	116
A.1.1 Visión general de mTeachPlan	116
A.1.2 Sincronización	117
A.1.3 Planificación de cursos	117
A.1.4 Planificación de sesiones	118
A.1.5 Gestión de preguntas y conceptos	120
A.1.6 Resultados	120
A.2 mClassTest: Manual del alumno	121
A.2.1 Iniciar sesión	121

A.2.2 Suscribirse a una sesión.....	121
A.2.3 Responder preguntas .....	122
<b>Apéndice B: Manual de mTeach Cloud Datastore .....</b>	<b>123</b>
B.1 Introducción.....	124
B.1.1 Propósito del documento .....	124
B.1.2 Histórico de versiones .....	124
B.1.3 Documentación de referencia .....	124
B.1.4 Datos de acceso al API .....	124
B.2 Descripción del modelo de datos.....	125
B.2.1 Claves, Integridad referencial y otras consideraciones .....	126
B.3 Introducción a Dropbox Datastore .....	126
B.3.1 Tipos de datos soportados .....	126
B.3.2 Organización del almacenamiento .....	126
B.4 Descripción del almacén de mTeach Cloud Datastore.....	127
B.4.1 Tabla courses .....	127
B.4.2 Tabla sessions.....	127
B.4.3 Tabla concepts .....	127
B.4.4 Tabla tweets .....	128
B.4.5 Tabla questions .....	128
B.4.6 Tabla responses .....	128

## Tabla de ilustraciones

Ilustración 1: Proceso de una asignatura .....	28
Ilustración 2: Interfaz de usuario de SRS .....	30
Ilustración 3: Interfaz de usuario de Blackboard en iPhone .....	31
Ilustración 4: Interfaz principal de Blackboard en iPad .....	31
Ilustración 5: Interfaz de Moodle .....	32
Ilustración 6: Distribución de la cuota de mercado en EEUU de los distintos sistemas operativos móviles en el tercer trimestre de 2012 .....	33
Ilustración 7: Capas del sistema operativo iOS .....	34
Ilustración 8: Entorno de desarrollo XCode 5. (Fuente: Apple Developer). .....	35
Ilustración 9: Ejemplo de manifiesto de gestión de dependencias con CocoaPods con múltiples módulos. ....	36
Ilustración 10: Diagrama de proceso de Scrum (Fuente: Mountain Goat Software) .....	38
Ilustración 11: Tablero Kanban (Fuente: Kanbanblog.com) .....	39
Ilustración 12: Funcionamiento de Kanban (Fuente: Kanbanblog.com) .....	39
Ilustración 13: Ciclo de vida de miSEL-sdp [17] .....	40
Ilustración 14: Diagrama de casos de uso para el actor Profesor .....	54
Ilustración 15: Diagrama de casos de uso para el actor Alumno .....	55
Ilustración 16: Primera alternativa arquitectónica .....	60
Ilustración 17: Segunda alternativa arquitectónica .....	61
Ilustración 18: Tercera alternativa arquitectónica .....	62
Ilustración 19: Diagrama de componentes .....	63
Ilustración 20: Detalle del componente 'Gestor de Conceptos' .....	64
Ilustración 21: Detalle del componente 'Gestor de sesiones' .....	65
Ilustración 22: Detalle de componente 'Gestor de preguntas' .....	66
Ilustración 23: Detalle del componente 'Gestor de cursos' .....	67
Ilustración 24: Detalle del componente 'Visor de estadísticas' .....	68
Ilustración 25: Detalle del componente 'Gestor de acceso a datos' .....	69
Ilustración 26: Controlador encargado del inicio de sesión de un cliente .....	70
Ilustración 27: Detalle del componente 'Navegador de sesiones' .....	71
Ilustración 28: Detalle del componente 'Inbox de preguntas' .....	72
Ilustración 29: Detalle de la librería de abstracción del API REST para iOS .....	73
Ilustración 30: Diagrama de secuencia 'Gestionar Cursos' .....	84
Ilustración 31: Diagrama de secuencia 'Gestionar Sesiones' .....	86
Ilustración 32: Diagrama de secuencia 'Gestionar Preguntas' .....	88
Ilustración 33: Diagrama de secuencia 'Gestionar Conceptos' .....	90
Ilustración 34: Diagrama de secuencia 'Gestionar Tweets' .....	92
Ilustración 35: Diagrama de secuencia 'Consultar estadísticas' .....	93
Ilustración 36: Diagrama de secuencia 'Suscribirse a sesión' .....	94
Ilustración 37: Imagen preliminar de la versión iOS 7 .....	111
Ilustración 38: Menú principal de mTeachPlan .....	116
Ilustración 39: Enlace de la aplicación con Dropbox .....	117
Ilustración 40: Pantalla de creación de cursos .....	117
Ilustración 41: Pantalla de edición de cursos .....	118
Ilustración 42: Pantalla de creación de sesiones .....	118
Ilustración 43: Contenidos de una sesión .....	119
Ilustración 44: Pantalla de creación de conceptos. ....	120
Ilustración 45: Pantalla de inicio de sesión de mClassTest .....	121
Ilustración 46: Ejemplo de suscripción a sesiones .....	121
Ilustración 47: Sesiones suscritas y bandeja de entrada de preguntas. ....	122
Ilustración 48: Pantalla de respuesta a preguntas .....	122
Ilustración 49: Esquema del modelo de datos empleado en iOS .....	125

## Índice de tablas

Tabla 1: Tabla comparativa de características entre aplicaciones	32
Tabla 2: Ejemplo de backlog con historias de usuario	44
Tabla 3: Product backlog	45
Tabla 4: Plantilla para la definición de requisitos	46
Tabla 5: Requisito RF-001 — Crear curso	46
Tabla 6: Requisito RF-002 — Modificar curso	47
Tabla 7: Requisito RF-003 — Borrar curso	47
Tabla 8: Requisito RF-004 — Crear sesión	47
Tabla 9: Requisito RF-005 — Modificar sesión	47
Tabla 10: Requisito RF-006 — Borrar sesión	48
Tabla 11: Requisito RF-007 — Crear concepto	48
Tabla 12: Requisito RF-008 — Modificar concepto	48
Tabla 13: Requisito RF-009 — Borrar concepto	48
Tabla 14: Requisito RF-010 — Crear pregunta	49
Tabla 15: Requisito RF-011 — Modificar pregunta	49
Tabla 16: Requisito RF-012 — Borrar pregunta	49
Tabla 17: Requisito RF-013 — Crear tweet	49
Tabla 18: Requisito RF-014 — Borrar tweet	49
Tabla 19: Requisito RF-015 — Consultar estadísticas	50
Tabla 20: Requisito RF-016 — Sincronización <i>cloud</i>	50
Tabla 21: Requisito RF-017 — Identificación con Twitter	50
Tabla 22: Requisito RF-018 — Suscribirse a una sesión	50
Tabla 23: Requisito RF-019 — Recibir preguntas	50
Tabla 24: Requisito RF-020 — Responder preguntas	51
Tabla 25: Requisito RF-021 — Listar preguntas recibidas	51
Tabla 26: Requisito RNF-001 — Dispositivos soportados	51
Tabla 27: Requisito RNF-002 — Versión mínima del sistema operativo iOS	51
Tabla 28: Requisito RNF-003 — Interfaz adaptativa	51
Tabla 29: Requisito RNF-004 — Uso del patrón de interfaz <i>Pull to refresh</i>	52
Tabla 30: Requisito RNF-005 — Seguimiento de las <i>Human Interface Guidelines</i> de Apple	52
Tabla 31: Matriz de trazabilidad requisitos - historias de usuario	53
Tabla 32: Caso de uso CU-001 — Gestionar cursos	55
Tabla 33: Caso de uso CU-002 — Gestionar sesiones	55
Tabla 34: Caso de uso CU-003 — Gestionar tweets	55
Tabla 35: Caso de uso CU-004 — Gestionar preguntas	56
Tabla 36: Caso de uso CU-005 — Gestionar conceptos	56
Tabla 37: Caso de uso CU-006 — Hacer <i>check in</i>	56
Tabla 38: Caso de uso CU-007 — Ver sesiones abiertas	57
Tabla 39: Caso de uso CU-008 — Suscribirse a sesión	57
Tabla 40: Caso de uso CU-009 — Responder preguntas	57
Tabla 41: Caso de uso CU-009 — Responder preguntas	57
Tabla 42: Matriz de trazabilidad requisitos - casos de uso	58
Tabla 43: Datos de llamada del método 'Crear sesión'	74
Tabla 44: Códigos de respuesta del método 'Crear sesión'	74
Tabla 45: Datos de llamada del método 'Listar sesiones abiertas'	74
Tabla 46: Códigos de respuesta del método 'Listar sesiones abiertas'	74
Tabla 47: Datos de llamada del método 'Consultar sesión'	75
Tabla 48: Códigos de respuesta del método 'Consultar sesión'	75
Tabla 49: Datos de llamada del método 'Finalizar sesión'	75
Tabla 50: Códigos de respuesta del método 'Finalizar sesión'	76
Tabla 51: Datos de llamada del método 'Eliminar sesión'	76
Tabla 52: Códigos de respuesta del método 'Eliminar sesión'	76
Tabla 53: Datos de llamada del método 'Eliminar sesión'	76
Tabla 54: Códigos de respuesta del método 'Eliminar sesión'	76
Tabla 55: Datos de llamada del método 'Check-in'	77

Tabla 56: Códigos de respuesta del método 'Check-in'	77
Tabla 57: Datos de llamada del método 'Suscribirse a una sesión'	77
Tabla 58: Códigos de respuesta del método 'Suscribirse a una sesión'	77
Tabla 59: Datos de llamada del método 'Darse de baja de una sesión'	78
Tabla 60: Códigos de respuesta del método 'Darse de baja de una sesión'	78
Tabla 61: Datos de llamada del método 'Listar preguntas de un alumno'	78
Tabla 62: Códigos de respuesta del método 'Listar preguntas de un alumno'	78
Tabla 63: Datos de llamada del método 'Obtener un alumno'	79
Tabla 64: Códigos de respuesta del método 'Obtener un alumno'	79
Tabla 65: Datos de llamada del método 'Borrar un alumno'	79
Tabla 66: Códigos de respuesta del método 'Borrar un alumno'	80
Tabla 67: Datos de llamada del método 'Crear pregunta'	80
Tabla 68: Códigos de respuesta del método 'Crear pregunta'	81
Tabla 69: Datos de llamada del método 'Obtener pregunta'	81
Tabla 70: Códigos de respuesta del método 'Obtener pregunta'	81
Tabla 71: Datos de llamada del método 'Borrar pregunta'	81
Tabla 72: Códigos de respuesta del método 'Borrar pregunta'	82
Tabla 73: Datos de llamada del método 'Responder pregunta'	82
Tabla 74: Códigos de respuesta del método 'Responder pregunta'	82
Tabla 75: Plantilla de ejemplo para la definición de pruebas	96
Tabla 76: Prueba PS-001 — Crear curso	96
Tabla 77: Prueba PS-002 — Modificar curso	97
Tabla 78: Prueba PS-003 — Borrar curso	97
Tabla 79: Prueba PS-004 — Crear sesión	97
Tabla 80: Prueba PS-005 — Modificación de una sesión	97
Tabla 81: Prueba PS-003 — Borrado de una sesión	98
Tabla 82: Prueba PS-007 — Crear un concepto	98
Tabla 83: Prueba PS-008 — Modificación de un concepto	98
Tabla 84: Prueba PS-009 — Borrado de un concepto	98
Tabla 85: Prueba PS-004 — Crear una pregunta	99
Tabla 86: Prueba PS-011 — Modificación de una pregunta	99
Tabla 87: Prueba PS-012 — Borrado de una pregunta	99
Tabla 88: Prueba PS-013 — Crear un tweet	100
Tabla 89: Prueba PS-014 — Borrar un tweet	100
Tabla 90: Prueba PS-015 — Consulta de estadísticas	100
Tabla 91: Prueba PS-016 — Sincronización	101
Tabla 92: Prueba PS-017 — <i>Check-in</i> con Twitter	101
Tabla 93: Prueba PS-018 — Suscribirse a una sesión	101
Tabla 94: Prueba PS-019 — Recibir preguntas	102
Tabla 95: Prueba PS-020 — Responder preguntas	102
Tabla 96: Matriz de trazabilidad pruebas de sistema – requisitos funcionales	103
Tabla 97: Costes de personal	106
Tabla 98: Costes de materiales	107
Tabla 99: Otros costes	107
Tabla 100: Coste total del proyecto	107
Tabla 101: Histórico de versiones	124
Tabla 102: Tabla courses	127
Tabla 103: Tabla sessions	127
Tabla 104: Tabla concepts	127
Tabla 105: Tabla tweets	128
Tabla 106: Tabla questions	128
Tabla 107: Tabla responses	128

---

# Capítulo 1: Introducción

---

## 1.1 Contexto

---

Con el lanzamiento de las dos principales redes sociales —Facebook y Twitter— en 2005 y 2006 y su rápida adopción por parte de muchos estudiantes se abrió un nuevo abanico de posibilidades de comunicación y colaboración entre docentes y alumnos. El poder comunicativo de dichas plataformas sociales aumentaba día tras día y de modo directamente proporcional al número de estudiantes que se incorporaban y al creciente tiempo de uso diario.

En el 2007, con la aparición del iPhone, en 2010 con la aparición del iPad y finalmente la aparición de *smartphones* y tabletas basadas en Android y con un coste menor supuso la popularización de este tipo de dispositivos hasta el punto que en nuestros días los estudiantes suelen disponer de al menos uno de estos dos tipos de dispositivos móviles, lo cual vuelve a abrir nuevas posibilidades educativas.

La combinación de las redes sociales con los dispositivos móviles supone un salto tecnológico en la forma de educar fusionando la capacidad comunicativa e inmediatez con la posibilidad de aprender allá donde se esté se rompen los límites clásicos del aula propiciando escenarios antes imposibles.

Imaginemos la siguiente situación: un profesor imparte una clase un viernes pero sin advertirlo comete un error en su explicación y la clase finaliza. Horas más tarde el profesor repara en su error y lanza un tweet en el que se corrige a sí mismo y sus alumnos lo reciben inmediatamente, por ejemplo en el autobús de vuelta a casa donde un alumno aprovecha el tiempo del viaje para recapitular lo visto durante la clase, ahora con la información libre de errores, optimizando así su tiempo.

Puesto que la comunicación es bidireccional puede suceder lo contrario, el alumno que viaja en el autobús puede advertir el error o exponer sus dudas abriendo así un debate con el profesor y el resto de sus compañeros quienes pueden responderle o poner de manifiesto que algo no ha quedado claro o no encaja sin tener que esperar a la siguiente sesión de clase que puede ser días más tarde y en la que ya nadie recordará que hubo un problema en la clase anterior fomentando la aparición de lagunas de conocimiento y el arraigo de errores conceptuales disminuyendo la calidad del aprendizaje del alumno.

El uso de tecnologías móviles para el aprendizaje combinado con redes sociales tiene las ventajas comentadas anteriormente, pero un abuso de dichas tecnologías también puede acarrear consecuencias negativas tales como la desviación del foco de atención del estudiante durante la propia clase, el hastío por un exceso de publicaciones, estudiar solamente aquello que se publique en los medios sociales y no profundizar en los materiales de la asignatura por considerarlo el material más relevante, etc.

Aunque se ha comentado que las tecnologías móviles y sociales están muy extendidas no todo el mundo posee una cuenta en una red social o un dispositivo móvil, por ello el uso de estas tecnologías debe considerarse un apoyo y una mejora a las técnicas docentes actuales pero no un reemplazo de las mismas ya que de ser así se produciría una injusta brecha entre los estudiantes.

Como se puede apreciar, a través del empleo de estas tecnologías el aprendizaje resulta mucho más atractivo e interactivo para el alumno, que ahora puede involucrarse más en el desarrollo de una asignatura dentro y fuera del aula y en cualquier momento y lugar.



## 1.2 Problema

Tradicionalmente los docentes se han enfrentado a la problemática que supone no poder conocer en el momento la marcha de las clases y si sus alumnos están siguiendo correctamente el progreso de las mismas, mantienen la atención y aprenden.

Una solución parcial al problema siempre han sido los ejercicios realizados durante una clase, bien sean ejercicios tipo test, bien ejercicios mucho más elaborados que el profesor tiene que evaluar tras la clase para no perder el tiempo del que dispone para la enseñanza. El profesor por lo tanto sólo puede obtener información a posteriori del resultado final de una clase—con un trabajo adicional de evaluación—y sólo puede hacer modificaciones de sus enseñanzas de cara a la próxima sesión en la que probablemente se traten otros conceptos, pudiendo quedar alumnos descolgados con dudas o lagunas.

Los alumnos también se ven en el problema de verse constantemente a prueba y estudian no para aprender sino para aprobar haciendo que el profesor obtenga resultados poco concluyentes sobre el progreso de sus pupilos y un *feedback* de escaso valor a la hora de mejorar la manera de impartir las clases.

El problema por lo tanto tiene dos vertientes, la de los profesores y la de los alumnos.

### 1.2.1 Profesor

El profesor busca obtener datos del progreso de su alumnado de un modo sencillo y rápido, sin la necesidad de elaborar materiales adicionales que requieran distribución, como por ejemplo fotocopias de cuestionarios que hay que distribuir, cumplimentar y posteriormente recoger dedicando tiempo a ello.

Los profesores también suelen disponer de poco tiempo para poder preparar materiales adicionales a la clase por lo tanto la manera de seguir el progreso del alumnado debe poder prepararse muy rápido, aprovechando cualquier hueco de tiempo disponible como por ejemplo los trayectos de desplazamiento en transporte público o entre clases.

### 1.2.2 Alumno

El alumno necesita poder seguir la clase de una manera en la que pueda obtener un aprovechamiento máximo de cada sesión, sabe que su profesor necesita información para dirigir la clase y que todo el mundo aprenda lo más posible pero es reacio a proporcionarla, por ejemplo cuando el profesor lanza una pregunta a la audiencia y nadie responde por desconocimiento o timidez o cuando siempre lo hacen los mismos y el profesor acaba adaptando la clase a una minoría de la audiencia.

El alumno por tanto necesita un medio de poder proporcionar dicha información con poco esfuerzo y con inmediatez, pero sin verse sometido a la presión que un ejercicio evaluable tradicional conlleva. El alumno debe poder sentirse parte activa de la clase y verse involucrado en el devenir de la misma.

### 1.2.3 Claves del problema

- Los profesores necesitan conocer sobre la marcha el desarrollo de una clase y poder preparar el material necesario para este proceso en cualquier momento y lugar.
- Se busca perder el menor tiempo posible sondeando a los alumnos.
- Los alumnos deben formar parte activa de la clase para mantener la atención, pero normalmente son reacios a ello salvo que se les proporcione un instrumento a través del cual hacer esto de manera inconsciente [1].

---

## 1.3 Motivación del proyecto

Del advenimiento de la generación actual de dispositivos móviles como teléfonos o tabletas, de las redes sociales, y de su omnipresencia en las aulas surge de manera natural la siguiente pregunta. ¿Cómo se podría aprovechar la gran difusión de estos dispositivos y servicios en las aulas y explotar los nuevos recursos y posibilidades docentes que ofrecen?

De la anterior pregunta se extrae la motivación de este proyecto consistente en:

- Proporcionar una serie de herramientas a los profesores para **preparar e impartir** sus clases, que funcionen en **sus dispositivos móviles** y que permitan completar ambas tareas con un esfuerzo mínimo añadido sobre el proceso tradicional.
- Proporcionar a los alumnos un **método** para **interactuar con la clase** a través de las **redes sociales** y a través de una herramienta destinada a **sus dispositivos personales**.
- Enlazar a profesores y alumnos a través de las nuevas tecnologías de modo que puedan cooperar fácilmente.
- **Sentar las bases para el desarrollo y la interoperabilidad** de un conjunto multiplataforma de aplicaciones móviles de soporte a la docencia de manera que juntas formen un entorno móvil de aprendizaje personal completo.

---

## 1.4 Objetivos del proyecto

El objetivo de este proyecto es desarrollar un conjunto de utilidades y servicios que permitan la preparación, el planteamiento, resolución y evaluación de cuestiones de tipo test a través de dispositivos móviles basados en el sistema operativo iOS, integradas en una suite multiplataforma que conforme un sistema completo de *Mobile Classroom Environment*. Este proyecto juega un papel fundamental en dicho sistema puesto que debe proporcionar los servicios necesarios sobre los que se asentarán el resto de módulos desarrollados a través de una serie de proyectos paralelos a este<sup>1</sup> y que una vez completos formarán un sistema *Personal Learning Environment* completo.

La consecución del objetivo global está sujeta al cumplimiento de los siguientes objetivos parciales

- **Creación de una aplicación iOS para el profesor** que permita gestionar sus cursos, las sesiones de cada curso, los conceptos que se van a tratar en la sesión y las preguntas sobre cada concepto que se van a plantear al alumnado.
- **Creación de una aplicación iOS para el alumno** que permita recibir y responder las preguntas que planteen los profesores durante las sesiones.
- **Enlazar las aplicaciones de alumnos y profesor** de tal forma que el alumno pueda recibir las preguntas y el profesor recibir las respuestas de cada alumno.
- **Enlazar las aplicaciones del profesor** tal que puedan compartir datos pero cada una sirva para su propósito específico, una para preparar sesiones y otra para impartirlas.
- **Proporcionar los servicios necesarios** sobre los que puedan asentarse el resto de módulos y aplicaciones del sistema de *Personal Learning Environment* e interoperar de forma cómoda y transparente para los usuarios.

---

<sup>1</sup> Este proyecto se encuadra dentro de un conjunto de aplicaciones de apoyo a la docencia desarrolladas por el grupo de investigación miSEL del *Software Engineering Lab* de la Universidad Carlos III de Madrid. Dentro de dicho paquete de aplicaciones se enmarcan proyectos como [7], [24], [26] y [25]

---

## 1.5 Método de resolución

Para la resolución del problema se ha optado por un enfoque iterativo basado en prototipos ya que desde un principio el conjunto de requisitos no ha estado completamente definido y se ha visto sujeto a cambios durante todo el proceso de desarrollo del proyecto.

El modo de realizar las distintas iteraciones de este proyecto se basó en las metodologías ágiles comentadas en la sección *Desarrollo ágil de software* del capítulo *Estado del arte* del presente documento.

Dado que el proceso de desarrollo era un proceso ágil, al inicio de cada iteración se establecía qué nuevas funcionalidades debían ser implementadas y se implementaban en ciclos de dos a tres semanas. Una vez implementadas, se validaban a través de una demostración. Debido a la naturaleza ágil del proceso, durante cada una de estas iteraciones podían definirse nuevos objetivos y requisitos, redefinir los ya existentes o eliminarlos. Por lo tanto había que estar preparado para los cambios que pudiesen surgir en cualquier momento.

A través de la demostración del prototipo se establecían los objetivos a completar de cara a la siguiente iteración hasta que se llegó al punto en el que todos los objetivos del proyecto quedaron conseguidos y se realizó la congelación de una versión funcional de la solución sobre la cual ya sólo se haría mantenimiento y corrección de errores y no se añadirían nuevas características.

Para la resolución del problema no fue necesaria una fase previa de aprendizaje del lenguaje o de la plataforma debido a que ya se contaba con experiencia previa sobre la misma, aprovechando el tiempo que se habría invertido en dicho proceso en construir un producto de mayor calidad con la ayuda de la experiencia previa con la que se contaba.

## 1.6 Definiciones, acrónimos y abreviaturas

En esta sección se definirán todos aquellos términos, acrónimos y abreviaturas que se empleen a lo largo de la extensión del presente documento y que el lector necesite conocer para comprender el texto en su totalidad.

### 1.6.1 Definiciones

- **E-Learning:** Anglicismo para usado para denominar el aprendizaje a través de medios electrónicos.
- **Framework:** Anglicismo usado para referirse a una plataforma software para el desarrollo de aplicaciones, productos y soluciones.

### 1.6.2 Acrónimos

- **ARC:** Acrónimo de *Automatic Reference Counting* o conteo automático de referencias, es una tecnología disponible en iOS que se encarga del manejo de la memoria en una aplicación a través del conteo automático de las referencias que se hacen a cada objeto.
- **GCC:** Acrónimo de *GNU Compiler Collection*.
- **GPL:** Acrónimo de *General Public License*, un tipo de licencia de software.
- **HTML:** Acrónimo de *HyperText Markup Language* o lenguaje de marcado de hipertexto.
- **IEEE:** Acrónimo de *Institute of Electrical and Electronics Engineers*, asociación profesional que se dedica a la creación, elaboración y documentación de estándares entre otras funciones
- **LMS:** Acrónimo de *Learning Management System* o sistema de gestión del aprendizaje
- **MVC:** Acrónimo de Modelo, Vista, Controlador.
- **P2P:** Acrónimo de *Peer to Peer*, en castellano ‘de igual a igual’.
- **POSIX:** Acrónimo de *Portable Operating System Interface*, es una familia de estándares creada por el IEEE para lograr compatibilidad entre sistemas operativos.
- **REST:** Acrónimo de *REpresentational State Transfer*.
- **RF:** Acrónimo de requisito funcional.
- **RNF:** Acrónimo de requisito no funcional.
- **UML:** Acrónimo de *Unified Modelling Language* o lenguaje de modelado unificado.
- **URL:** Acrónimo de *Universal Resource Locator* o localizador universal de recursos, también conocido como dirección web.
- **US:** Acrónimo de *User Story*, o historia de usuario.

### 1.6.3 Abreviaturas

- **N/A:** No aplicable.
- **SO:** Sistema Operativo.

---

## 1.7 Visión general del documento

En este apartado se explicará brevemente cómo se estructura toda la información que se va a ir presentando en los diversos capítulos.

En el primer capítulo, “Introducción”, se presentará el problema que este proyecto pretende resolver, la motivación para resolverlo, los objetivos que se han de cumplir para dar el problema por resuelto y qué método se va a emplear para abordar la solución de dicho problema.

El capítulo también incluye una relación de términos, acrónimos y abreviaturas con sus respectivas definiciones, las cuales ayudarán a la comprensión de los diferentes acrónimos que se encontrarán a lo largo de todo el documento.

En el segundo capítulo, “Estado del arte”, se realizará una exposición tanto de otras herramientas similares en el campo de *e-learning* como de las tecnologías que se han empleado para resolver el problema y que conforman el contexto de la solución desarrollada, asimismo se explicará la metodología empleada durante el proceso de desarrollo de la solución.

En el tercer capítulo, “Análisis del problema”, se efectuará la labor de analítica que definirá los requisitos, historias de usuario y cuantos productos adicionales que sean necesarios para conformar una especificación sólida del problema sobre la cual trabajar en el resto de fases del proceso de desarrollo.

En el cuarto capítulo, “Diseño de la solución” se procederá a definir la arquitectura de la solución y su apariencia, estructura y comportamiento. Esto se realizará a través de diagramas de componentes, diagramas de clases y diagramas de secuencia.

En el quinto capítulo, “Pruebas del sistema” se definirán las pruebas que debe pasar la solución para ser validada. Se realizará una especificación de las pruebas que deben implementarse para testar la validez de la solución.

En el sexto capítulo, “Presupuesto” se expondrán los costes asociados al desarrollo de proyecto.

En el séptimo y último capítulo, “Conclusiones y líneas futuras” se exponen las conclusiones obtenidas tras completar el proceso de desarrollo del proyecto así como futuras líneas de trabajo para la mejora y ampliación de la solución.



---

## Capítulo 2: Estado del arte

---

## 2.1 Introducción a los entornos de aprendizaje personal

Los entornos de aprendizaje personal o PLE, acrónimo inglés de *Personal Learning Environment* comprenden las herramientas y servicios que componen plataformas educativas que los estudiantes usan para su propio aprendizaje [2].

En un PLE intervienen —principalmente— dos clases de actores, los profesores y los alumnos. Los profesores preparan los contenidos de las clases y los alumnos los consumen, pero también generan nuevos materiales como producto de su proceso de aprendizaje que sirven para realimentar el proceso docente y que fomentan el intercambio de conocimientos con otros compañeros a través de las herramientas que el PLE proporciona a los usuarios.

Un PLE debe cubrir todo el proceso típico del curso de una asignatura, por ello debe proporcionar a los actores las herramientas apropiadas para cubrir cada una de las fases de dicho proceso.

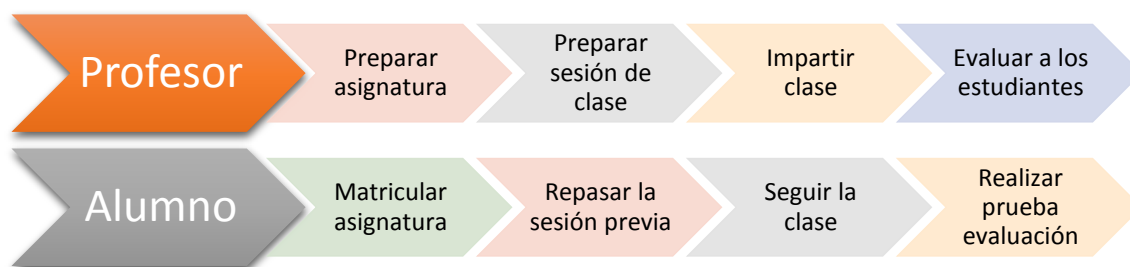


Ilustración 1: Proceso de una asignatura

Antes del inicio del curso, un profesor debe preparar ciertos contenidos de la asignatura, como su estructura, una planificación, una bibliografía y todo aquello que resulte relevante al docente ofrecer a su alumnado al inicio del curso. En esta fase el alumno lo que el alumno debe realizar son las gestiones pertinentes para poder seguir la asignatura a través del PLE.

En la siguiente fase, el profesor debe preparar la sesión de clase que va a impartir, los contenidos que desea emplear y todo tipo de recursos que vaya a necesitar durante la sesión de clase. El alumno por su parte en esta fase debe repasar la sesión previa de haberla y si es necesario para poder seguir adecuadamente la siguiente sesión.

En la tercera fase el profesor imparte la clase y pone en juego los recursos que ha preparado en la fase anterior, transparencias, preguntas a la clase, vídeos, publicaciones en redes sociales, etc. El PLE debe dar soporte al uso de estos recursos. El alumno en este punto debe seguir la clase y las explicaciones, el PLE en este punto debe ser un apoyo y no una distracción y debe ser usado con cautela para presentar información complementaria pero que no desvíe el foco de atención del estudiante.

Finalmente, los alumnos deben ser evaluados por sus profesores y el PLE también debe dar soporte a ello, recopilando las respuestas de los alumnos a las pruebas que los profesores planteen y ofreciendo al profesor una herramienta que permita evaluar (y en su caso, calificar) los conocimientos adquiridos por sus estudiantes completando así el proceso docente.



## 2.2 Análisis de aplicaciones relacionadas con los objetivos establecidos

En esta sección se procederá a hacer un análisis de aplicaciones existentes relacionadas con los objetivos del proyecto, si se ajustan a los mismos, si son viables y qué características poseen que pueden satisfacer los objetivos o divergen demasiado de los mismos.

Se ha de recordar que este proyecto es tan solo una parte de un sistema PLE completo que cubrirá todas las fases del proceso de una asignatura comentadas en *Introducción a los entornos de aprendizaje personal* y formado por distintos módulos desarrollados en proyectos independientes y que por tanto los objetivos van más allá de los expuestos en el apartado *Objetivos del proyecto*.

### 2.2.1 EduClick

EduClick es un sistema desarrollado por EduClick S.L que consiste en un sistema de mandos a distancia y receptores de señal para poder procesar las respuestas del alumnado a una pregunta que un profesor plantee durante una clase. Según el fabricante del sistema el funcionamiento es el siguiente [3]:

1. Definición previa de preguntas o cuestiones multimedia interactivas.
2. Mostrando una pregunta, el ponente activa el inicio de las preguntas.
3. Los participantes responden pulsando en su mando la tecla que corresponde a su respuesta.
4. Las respuestas se almacenan permitiendo mostrar un gráfico de resultados inmediatamente.
5. La sección de informes permite explotar la información almacenada sesión tras sesión.

La solución que plantea EduClick se ajusta bastante a los objetivos propuestos para este proyecto pero se trata de una solución propietaria muy difícil de integrar con el resto de módulos del PLE. También tiene el gran inconveniente de estar basada en hardware propietario —los mandos y receptores— hacen inviable su despliegue en un entorno universitario en el que hay muchos alumnos por clase y es necesario adquirir un elevado número de dispositivos del que además se desconoce su coste. EduClick también ofrece una aplicación para dispositivos móviles pero con una funcionalidad reducida al sistema de mandos hardware.

Es de suponer, debido a la naturaleza propietaria de la solución que cada mando o instalación de aplicación móvil conllevará la adquisición de una licencia de usuario, aparte de la plataforma como tal y un plan de mantenimiento elevando posiblemente hasta niveles prohibitivos el coste total de la propiedad de un sistema como EduClick en una institución como una universidad en la que simultáneamente se celebran muchas clases a las que deben asistir grupos grandes de estudiantes.

### 2.2.2 Student Response System — SRS

*Student Response System*, más conocido por sus siglas SRS, es un sistema desarrollado por el Trondheim University College (HiST – Høgskolen i Sør-Trøndelag) de Noruega a través del proyecto EduMecca [1].

SRS fue desarrollado con un objetivo en mente similar al de este proyecto, poder hacer preguntas al alumnado durante las clases y que los estudiantes puedan responder desde sus propios dispositivos conectados a internet. SRS está desarrollado como una aplicación web, de modo que todo dispositivo con conexión a internet y un navegador web pueda acceder a la aplicación, bien sea un ordenador personal o un dispositivo móvil [4].

Aunque la naturaleza independiente de la plataforma de SRS es una ventaja carece de funcionalidades avanzadas como las que se requieren en este proyecto, tales como permitir a un profesor preparar todo el material de sus clases allá donde esté y mantenerlo sincronizado entre todos sus dispositivos, integración con redes sociales o de combinar materiales como preguntas con diapositivas concretas de una presentación.

SRS requiere disponible un ordenador de escritorio en el aula para que el profesor pueda manejar el lanzamiento de preguntas lo cual impide el uso de este sistema en clases de campo o en lugares que difieran de un aula tradicional.

SRS está únicamente disponible por el momento para un uso gratuito limitado de evaluación a pequeña escala bajo solicitud al Sør-Trøndelag University College [5] y no está preparado para un despliegue a mayor escala. Tampoco hay detalle de futuros precios por el uso del sistema ni visos de facilidades para la integración dentro de un sistema PLE completo.

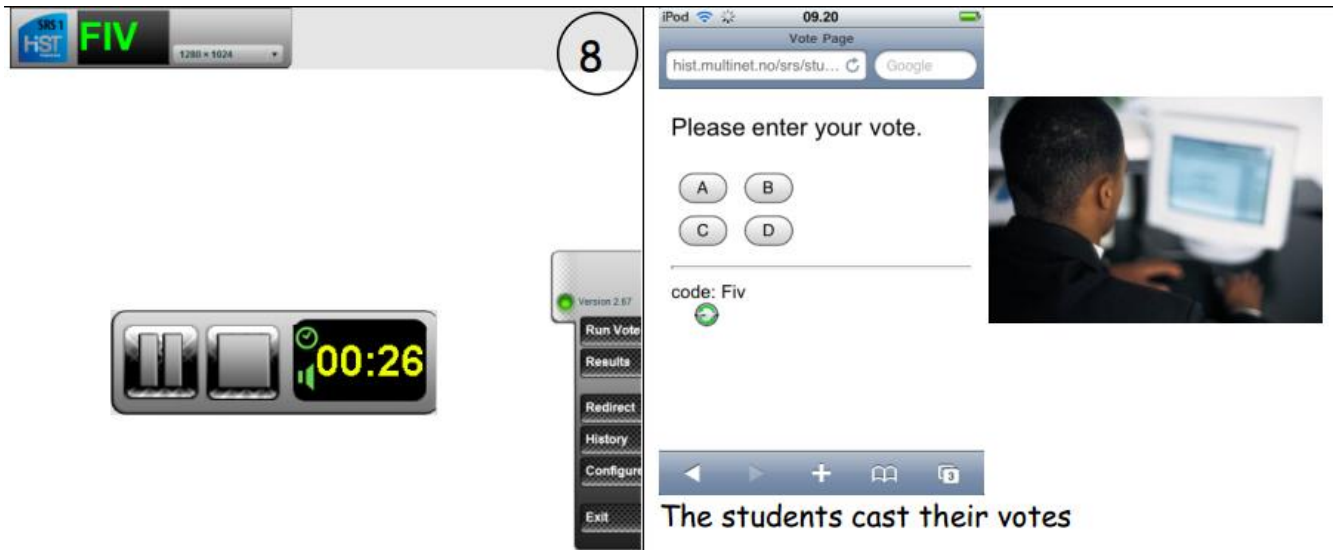


Ilustración 2: Interfaz de usuario de SRS

SRS carece de notificaciones a los dispositivos suscritos a la sesión de clase con lo cual, los alumnos deben tener siempre activos sus dispositivos y estar alerta a los mismos con el consiguiente consumo de batería y distracción para el alumno.

### 2.2.3 Blackboard

Blackboard es una solución LMS propietaria que tanto profesores como alumnos pueden emplear para acceder a sus diferentes cursos y contenidos de estos de una manera atractiva e intuitiva. Entre algunas de sus características se destacan el acceso a los documentos de un curso en múltiples formatos, consulta de anuncios de noticias importantes, creación de hilos de discusión, carga de archivos multimedia, creación de elementos dentro de un curso, etc.

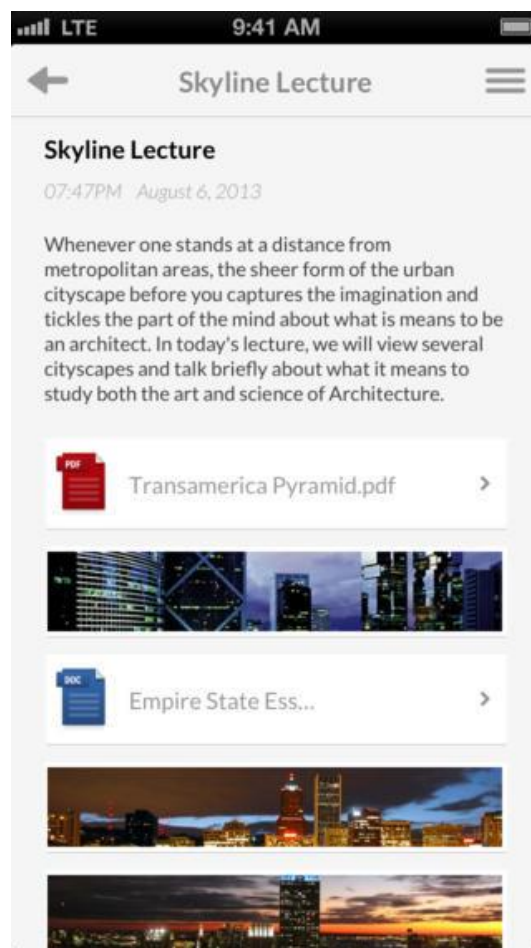


Ilustración 3: Interfaz de usuario de Blackboard en iPhone

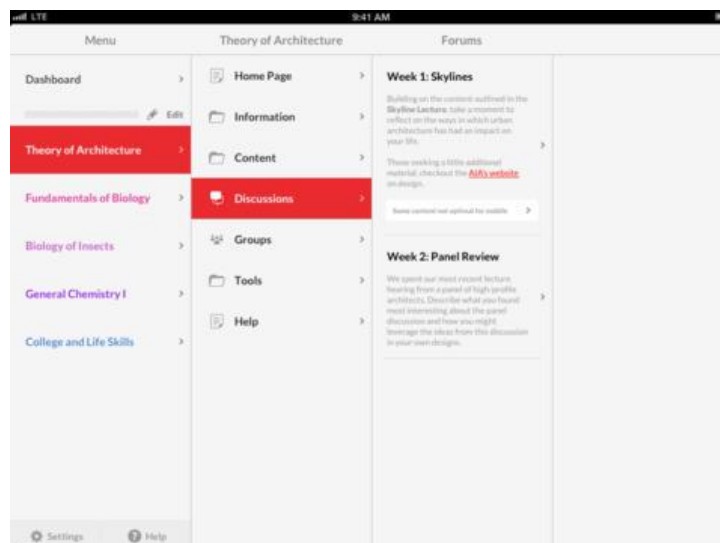


Ilustración 4: Interfaz principal de Blackboard en iPad

## 2.2.4 Moodle

Basada en tecnología HTML, esta aplicación permite la conexión a un servidor web donde se haya instalado la plataforma. Esta herramienta busca unificar la gestión del aprendizaje y promover el mismo. Con tal fin, Moodle cuenta con algunas características, tales como una gran escalabilidad y facilidad de uso para administradores y estudiantes, que le permite adaptarse tanto a instituciones de educación primaria como a universidades con cientos de miles de estudiantes o la posibilidad de realizar cursos enteramente on-line y a distancia, proporcionando un lugar donde centralizar los recursos y hacerlos disponibles a través de la plataforma y proporcionar herramientas como foros, bases de datos, wikis, etc.

Moodle es otra herramienta de tipo LMS, gratuita, de código abierto y que facilita su integración en un sistema PLE



Ilustración 5: Interfaz de Moodle

## 2.2.5 Comparativa entre aplicaciones

En este apartado se va a mostrar una comparativa de qué aplicaciones pueden cubrir qué necesidades de las que se han planteado.

	Preparar asignatura	Preparar clase	Impartir clase	Evaluar	Integración móvil
EduClick	—	—	X	X	Parcial <sup>2</sup>
SRS	—	—	X	X	X
Blackboard	X	X	—	Parcial <sup>3</sup>	X
Moodle	X	X	—	X	Parcial <sup>4</sup>

Tabla 1: Tabla comparativa de características entre aplicaciones

<sup>2</sup> La versión para móviles de EduClick está muy limitada con respecto a la que hace uso de mandos físicos

<sup>3</sup> Blackboard no permite realizar preguntas aisladas en tiempo real, plantea tests completos.

<sup>4</sup> El soporte móvil de Moodle es a través de una *webapp* y no una aplicación nativa

## 2.3 Elección de la plataforma de desarrollo

Según se puede extraer del informe de Nielsen para el tercer trimestre de 2012 sobre la cuota de mercado de los distintos sistemas operativos móviles [6], la distribución de dicha cuota de mercado en los EEUU es de un 52% para el sistema operativo Android y de un 35% para el sistema operativo iOS.

### Top U.S. Smartphone Operating Systems by Market Share

Q3 2012, Nielsen Mobile Insights

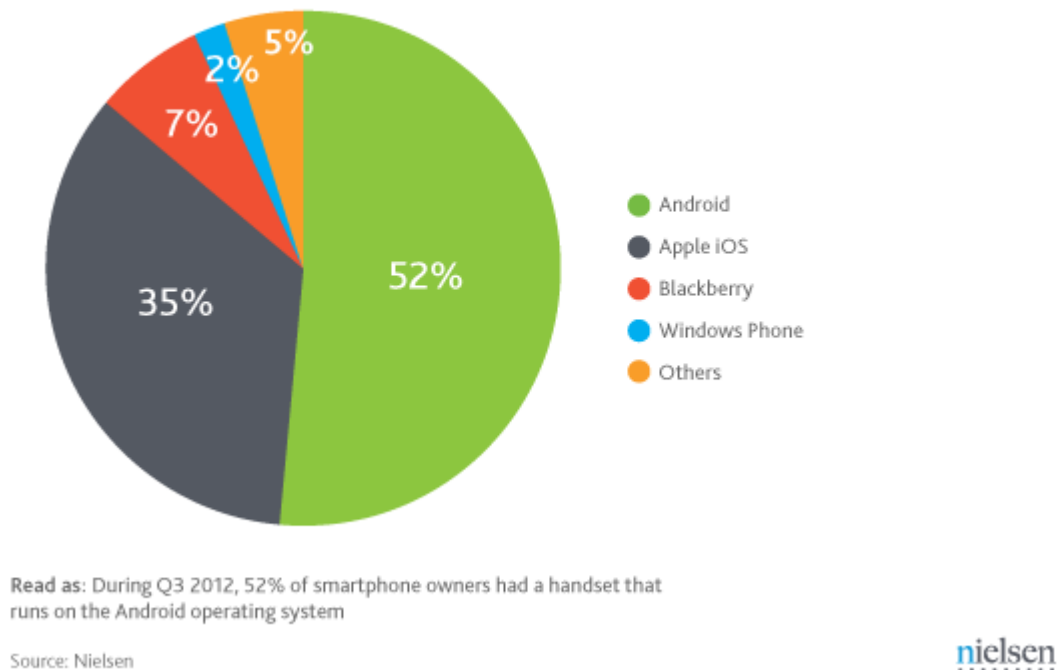


Ilustración 6: Distribución de la cuota de mercado en EEUU de los distintos sistemas operativos móviles en el tercer trimestre de 2012

Sin embargo, aunque la cuota de mercado de dispositivos Android es mayor, conviven multitud de versiones del sistema y tipos distintos de dispositivos con multitudes de diferentes tamaños de pantalla y capacidades convirtiéndola en una plataforma para la que es más difícil desarrollar.

Por el contrario, iOS proporciona una plataforma de desarrollo mucho menos fragmentada que acelera el desarrollo y despliegue de una aplicación. Aunque la cuota de mercado es menor que la de Android está lo suficientemente extendido como para lanzar una aplicación antes en esta plataforma, obtener datos significativos de uso que demuestren el éxito o fracaso de una idea y según los resultados obtenidos decidir si merece la pena invertir los recursos necesarios en llevar la idea a Android para así poder abarcar el 87% del mercado de dispositivos móviles.

Como este proyecto y su proyecto compañero [7] son las experiencias piloto en base a las que se desarrollarán futuros proyectos basados en estos para distintas plataformas se desarrollarán primero bajo la plataforma iOS para después decidir si han de ser portadas al resto de plataformas en base a los datos que se recaben de su uso.

## 2.4 El entorno de desarrollo iOS

iOS es el sistema operativo que emplean los dispositivos móviles de Apple, como los iPhone, iPod Touch o los iPad. El proceso de desarrollo para crear aplicaciones nativas para estos dispositivos se realiza a través de lo que se conoce como **iOS Software Development Kit (SDK)**.

El conjunto de herramientas conocido como iOS Software Development Kit (SDK) contiene las herramientas e interfaces necesarias para desarrollar, instalar, ejecutar y probar aplicaciones nativas que aparecen en la pantalla de inicio de un dispositivo iOS. Las aplicaciones nativas son construidas empleando los *frameworks* del sistema iOS y el lenguaje de programación Objective-C y se ejecutan directamente sobre iOS. [8]

### 2.4.1 Estructura de iOS

El sistema operativo iOS se estructura como una serie de cuatro capas que ofrecen distintos servicios al desarrollador de aplicaciones y que abstraen del hardware y de sus cambios futuros a las aplicaciones.

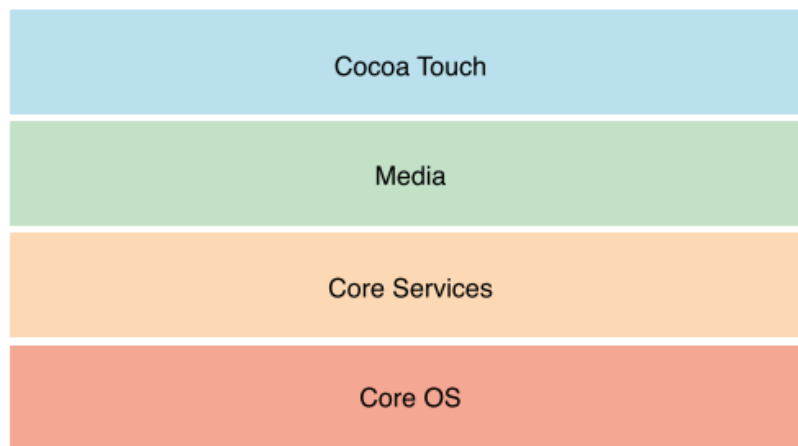


Ilustración 7: Capas del sistema operativo iOS

Desde el nivel más bajo al más alto se explicará brevemente qué servicios ofrece cada capa.

- **Core OS:** La capa *Core OS* ofrece los servicios más elementales del sistema como el soporte base para hilos (*POSIX Threads*), *sockets*, entrada/salida básica, reserva de memoria, etc.
- **Core Services:** La capa *Core Services* ofrece servicios más avanzados que abstraen aquellos que proporciona *Core OS*, tales como *Core Data* —un servicio de persistencia de datos basado en un grafo de entidades y relaciones—, *Core Foundation* que ofrece manejo de cadenas de texto, *URLs*, fechas, manejo automático de memoria a través de *ARC* o una abstracción para manejar hilos.
- **Media:** La capa *Media* se encarga de proporcionar servicios para el dibujado de elementos en pantalla, reproducción de sonidos, acceso a la biblioteca musical del dispositivo, etc. Se encarga de exponer todas las funciones multimedia del dispositivo a las aplicaciones que quieran consumirlas.
- **Cocoa Touch:** *Cocoa Touch* es la capa que proporciona se encarga de gestionar la interfaz de usuario. Provee al desarrollador de una serie de mecanismos y elementos estándar para construir las interfaces de usuario y ofrecer una experiencia de usuario consistente entre aplicaciones.



## 2.4.2 El lenguaje de programación Objective-C

Como lenguaje, Objective-C posee una larga historia. Fue creado en la compañía Stepstone a principios de los 80 por Brad Cox y Tom Love. A finales de los 80 fue licenciado por NeXT Computer Inc. para el desarrollo de los *frameworks* de NeXTStep que precedieron a Cocoa. NeXT extendió el lenguaje posteriormente de varias formas, por ejemplo, con la adición de los protocolos. [9]

Objective-C es un lenguaje de programación orientado a objetos que extiende la sintaxis del lenguaje C y por lo tanto le convierte en una extensión o superconjunto del mismo, es decir, todo programa escrito en lenguaje C es también un programa Objective-C —pero no al revés— lo cual permite portar código escrito en C ya existente sin apenas esfuerzo.

El desarrollo del lenguaje comenzó como un intento de añadir a C algunas de las capacidades de Smalltalk, como la orientación a objetos.

Posteriormente, cuando NeXT licenció el lenguaje a Stepstone lo implementó en el compilador GCC y fue liberado debido a las imposiciones de la licencia GPL, bajo la cual se licencia el compilador GCC y que obliga a liberar bajo la misma licencia las modificaciones que se hagan al propio GCC.

En el año 2006 Apple introdujo Objective-C 2.0, con nuevas características como soporte de arquitecturas de 64 bits o recolección de basura mejorada en el sistema operativo MacOS X<sup>5</sup>

## 2.4.3 El entorno de desarrollo integrado XCode

Cuando Apple compró a NeXT, aparte de Objective-C también compraron las herramientas de desarrollo que NeXT había desarrollado y entre ellas se encontraba Project Builder, el padre de XCode.

XCode se desarrolló a partir de Project Builder y la primera versión se lanzó al público en el año 2003. [10]

XCode es un entorno integrado de desarrollo moderno, con todas las funcionalidades que cabría esperar, tales como un editor de código fuente, autocompletado de código, un navegador de la estructura del proyecto, un compilador, un editor de interfaces gráficas o un depurador entre muchas otras. [11]



Ilustración 8: Entorno de desarrollo XCode 5. (Fuente: Apple Developer).

<sup>5</sup> En iOS el *runtime* de Objective-C carece de recolector de basura y el manejo del tiempo de vida de un objeto en memoria se realiza mediante conteo de referencias. [23]

## 2.4.4 La gestión de dependencias y CocoaPods: supervivencia en un mar de caos

En un proyecto software de cierta complejidad es habitual verse en la necesidad de recurrir a componentes de terceros, bien producidos dentro de la propia organización, bien licenciados a terceros o bien emplear componentes de código abierto que se adecúen a nuestras necesidades evitando reinventar la rueda en cada proyecto.

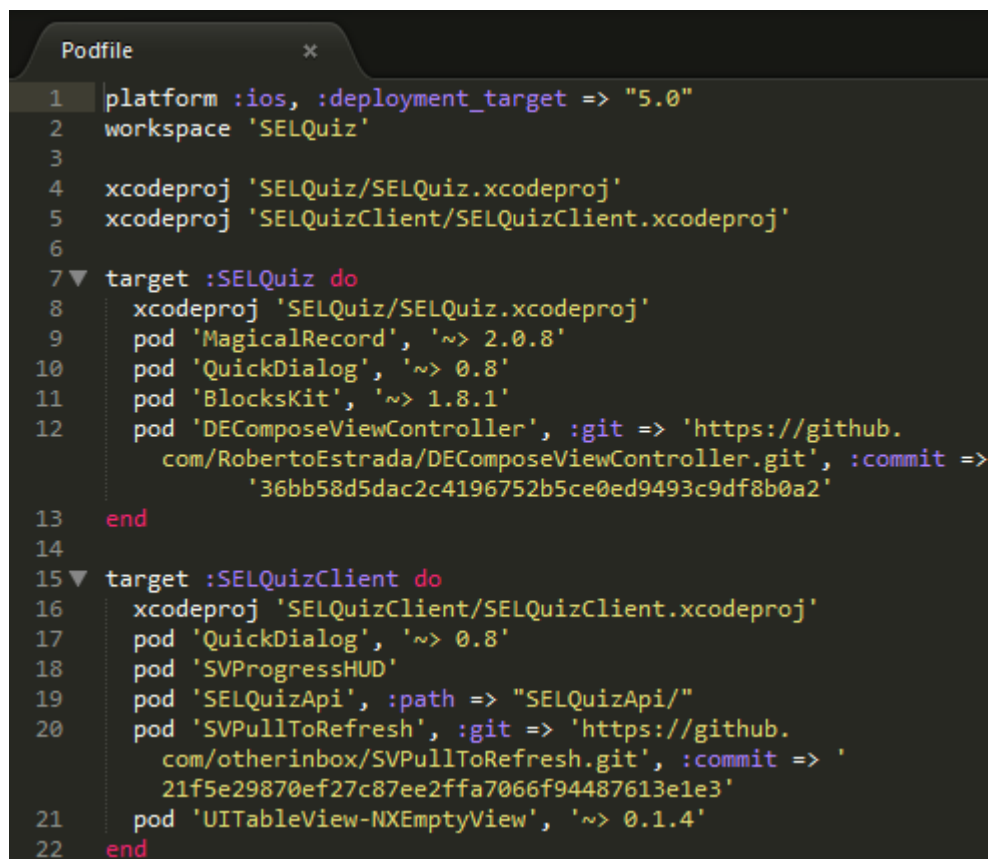
Dichos componentes externos de los que depende el proyecto son suelen tener cada uno su propia y peculiar manera de ser integrados en el proceso de construcción y su reemplazo por otros puede llegar a convertirse en un proceso tremendamente doloroso.

Para resolver este problema nació CocoaPods, un proyecto de código abierto que permite integrar en el proceso de construcción de un proyecto todas las dependencias que sean necesarias especificándolas a través de un fichero de texto plano. Asimismo las dependencias a ser integradas también deberán poseer un ‘manifiesto’ que especifique a CocoaPods cómo ser integradas en el proceso de construcción y si además poseen subdependencias que deban satisfacerse.

Los creadores de CocoaPods describen el proyecto de la siguiente forma.

*CocoaPods manages library dependencies for your Xcode project. You specify the dependencies for your project in one easy text file. CocoaPods resolves dependencies between libraries, fetches source code for the dependencies, and creates and maintains an Xcode workspace to build your project. Ultimately, the goal is to improve discoverability of, and engagement in, third party open-source libraries, by creating a more centralized ecosystem. [12]*

CocoaPods funciona creando un proyecto de tipo biblioteca estática que contiene el código compilado de las dependencias contra el que se enlaza el proyecto principal, también además permite especificar diferentes dependencias para cada uno de los módulos que componen un proyecto.



```

Podfile
1 platform :ios, :deployment_target => "5.0"
2 workspace 'SELQuiz'
3
4 xcodeproj 'SELQuiz/SELQuiz.xcodeproj'
5 xcodeproj 'SELQuizClient/SELQuizClient.xcodeproj'
6
7 target :SELQuiz do
8   xcodeproj 'SELQuiz/SELQuiz.xcodeproj'
9   pod 'MagicalRecord', '~> 2.0.8'
10  pod 'QuickDialog', '~> 0.8'
11  pod 'BlocksKit', '~> 1.8.1'
12  pod 'DEComposeViewController', :git => 'https://github.
    com/RobertoEstrada/DEComposeViewController.git', :commit =>
    '36bb58d5dac2c4196752b5ce0ed9493c9df8b0a2'
13 end
14
15 target :SELQuizClient do
16   xcodeproj 'SELQuizClient/SELQuizClient.xcodeproj'
17   pod 'QuickDialog', '~> 0.8'
18   pod 'SVProgressHUD'
19   pod 'SELQuizApi', :path => "SELQuizApi/"
20   pod 'SVPullToRefresh', :git => 'https://github.
    com/otherinbox/SVPullToRefresh.git', :commit => '
    21f5e29870ef27c87ee2ffa7066f94487613e1e3'
21   pod 'UITableView-NXEmptyView', '~> 0.1.4'
22 end
  
```

Ilustración 9: Ejemplo de manifiesto de gestión de dependencias con CocoaPods con múltiples módulos.



## 2.5 Desarrollo ágil de software

El desarrollo ágil de software nace como una alternativa basada en los procesos de desarrollo iterativos como alternativa a los tradicionales y lentos procesos de desarrollo tradicionales basados mayoritariamente en el modelo de cascada.

En el desarrollo ágil se hace hincapié en lo que lo que el equipo de desarrollo de un producto debe hacer: sacar adelante un proyecto en el menor tiempo y con la mayor calidad y valor añadido posible reduciendo al máximo todo lo superfluo de un proceso tradicional como malgastar tiempo en extensiva burocracia y documentos superfluos, productos que a un cliente le sirven de bien poco, puesto que lo que desea es una aplicación funcionando pronto y conforme a sus expectativas.

Los procesos ágiles buscan mantener motivado al equipo de desarrollo a través de la simplicidad y poniendo énfasis en las interacciones entre los individuos del equipo y del establecimiento de hitos en el desarrollo —a través de una planificación activa— [13] en los que se obtiene un producto entregable y que contiene un subconjunto de la funcionalidad total pero que al fin y al cabo funciona, proporcionando gran satisfacción personal a los miembros de un equipo tras cada hito. También logran este objetivo respondiendo ante el cambio, algo que en procesos tradicionales supone una absoluta frustración para un equipo en un proceso ágil es algo que ni de lejos resulta tan traumático.

En resumen, el desarrollo ágil de software persigue los siguientes puntos:

- Entregas frecuentes de versiones funcionales en el menor tiempo posible.
- Reducir al máximo la complejidad del proceso, lo importante es desarrollar no el papeleo.
- El progreso se mide según funcione más o menos de la aplicación
- ¿Cambios en el proyecto? No importa. Sabíamos que podían ocurrir y estamos preparados ante ello.

Existen diversas metodologías de desarrollo ágil que se emplean en la industria, tales como UP (*Unified Process*), SCRUM, XP (*Extreme Programming*) o Kanban por citar algunas de las más conocidas.

Todas estas metodologías de desarrollo asumen equipos de desarrollo de más o menos individuos pero, ¿qué ocurre si llevamos esta asunción al extremo? ¿Y si el equipo de desarrollo está compuesto por un solo miembro? El problema que surge no es muy grande, pues el proceso puede miniaturizarse sin demasiadas complicaciones eliminando o modificando elementos que en un equipo unipersonal no resultan adecuados.

### 2.5.1 Scrum

Scrum es una metodología para el desarrollo y mantenimiento de productos complejos mediante la cual las personas pueden atajar problemas de un modo productivo y creativo, generando productos de gran valor. [14]

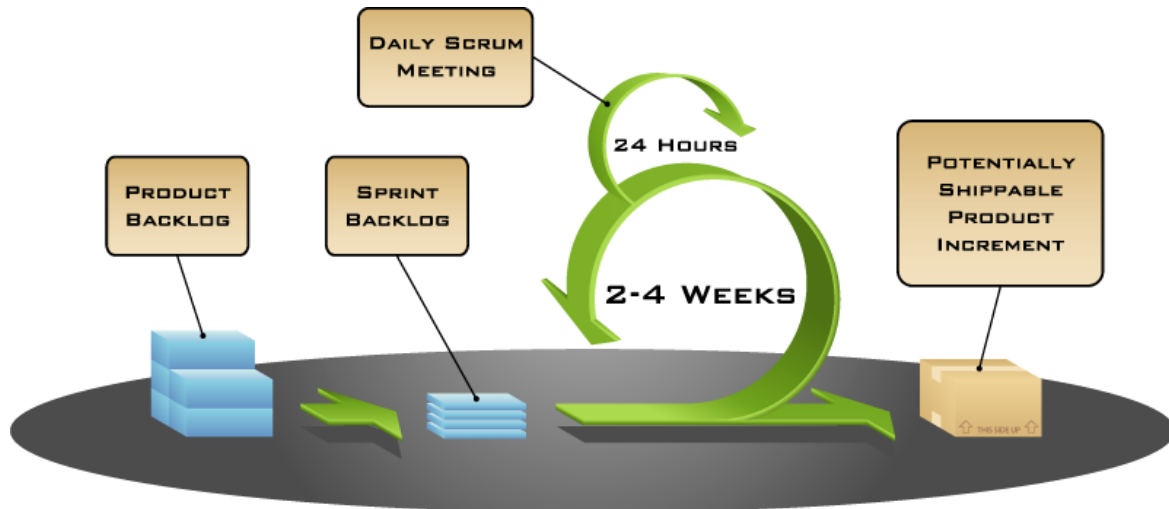
La palabra *scrum* proviene del mundo del deporte, en concreto del rugby. En el rugby un *scrum* es un agrupamiento donde ambos equipos se ponen frente a frente y pugnan —como equipo y no como individuos— para avanzar con la pelota a la meta del rival.

La metodología Scrum tiene como objetivo realizar entregas parciales a intervalos regulares y funciona muy bien en entornos complejos donde los requisitos están poco definidos o son bastante propensos al cambio.

Scrum se compone de los siguientes procesos y artefactos.

- **Product backlog:** también conocida como la pila del producto, es una pila donde se van poniendo las historias de usuario —una suerte de requisitos escritos en apenas unas líneas de un lenguaje similar al que emplearía un usuario— que tendrán que implementarse durante el desarrollo del proyecto.

- **Sprint:** una iteración del proceso de desarrollo, normalmente de una longitud de una a dos semanas.
- **Sprint backlog:** similar a la pila del producto pero limitada a la duración de un *sprint*. En dicha pila se deberán colocar las tareas que del *backlog* que se deseen realizar en el *sprint* actual.
- **Reunión diaria:** reunión que debe realizarse todos los días aproximadamente a la misma hora y durante un periodo de tiempo corto de unos quince minutos es lo ideal donde los miembros del equipo comentan brevemente qué hicieron el último día, qué van a hacer hoy y si tienen algún impedimento que lastre su avance. Esta reunión debe hacerse al inicio de la jornada laboral del equipo para establecer un contexto de trabajo para el día presente.



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Ilustración 10: Diagrama de proceso de Scrum (Fuente: Mountain Goat Software)

## 2.5.2 Kanban

La técnica Kanban sirve para manejar procesos de desarrollo de software de una manera altamente eficiente. Kanban recoge la idea subyacente en la técnica de producción industrial de Toyota, el famoso *Toyota Production System* o TPS. Aunque la producción de software es un proceso eminentemente creativo a diferencia de la producción masiva de automóviles los principios básicos de la técnica también aplican para el desarrollo de software. [15]

Kanban permite ver el proceso de producción de un proyecto a través de un tablero en el que se tiene una lista de tareas pendientes y las distintas fases de la cadena por las que tiene que pasar una tarea antes de considerarse completada.

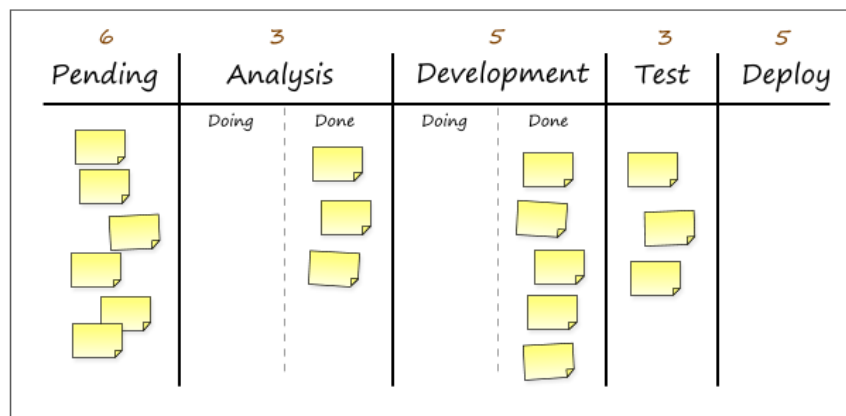


Ilustración 11: Tablero Kanban (Fuente: Kanbanblog.com)

Una de las grandes ventajas de Kanban es que gracias a su representación visual permite identificar rápidamente cuellos de botella en las distintas fases del proceso. Es decir, si empiezan a acumularse tarjetas (tareas) en una de las columnas (una fase) habría que preguntarse qué sucede en esa fase y qué puede hacerse por solucionarse para evitar el estancamiento de tareas y el estrangulamiento del proceso de producción.

Kanban funciona del siguiente modo, se toman tareas de la pila de tareas pendientes y se ponen en la primera fase, cuando se ha completado dicha fase para la tarea se mueve a la siguiente y así sucesivamente hasta llegar al final de la cadena.

A continuación un ejemplo ilustrado del proceso

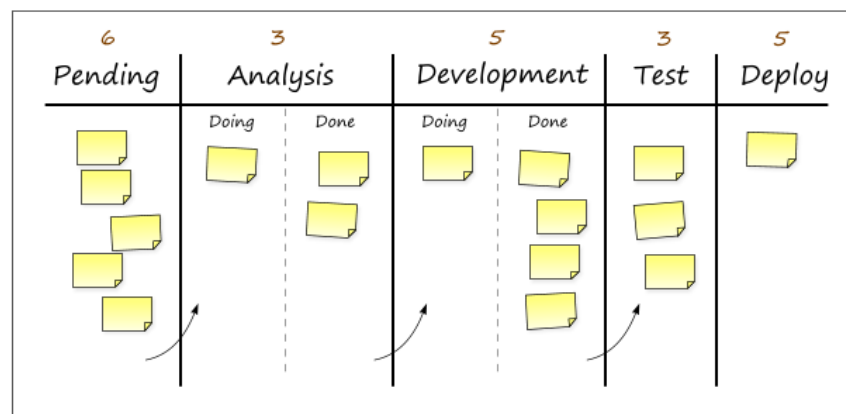


Ilustración 12: Funcionamiento de Kanban (Fuente: Kanbanblog.com)

### 2.5.3 Scrumban

Scrumban es una metodología —o más bien una mezcla de metodologías— en la que se combinan flexiblemente, según las necesidades de quien implemente la metodología, elementos de Scrum y de Kanban. Scrumban no es una metodología perfectamente definida y por eso puede amoldarse a circunstancias menos comunes tales como equipos de desarrollos unipersonales como es el caso de este proyecto.

Algunas de las principales características de Scrumban son [16]:

- No se requiere del concepto equipo.
- No se requieren sprints, es decir, no hay plazos fijos entre entregas.
- No se requiere una reunión diaria de equipo

- No se requiere una retrospectiva.

Si bien tanta laxitud puede parecer un paso atrás, en casos como el que nos ocupa es más bien un paso hacia delante, ya que de entrada no existe un equipo y es de todo punto absurdo realizar reuniones ya que el desarrollador conoce a cualquier hora del día el estado de desarrollo del proyecto, sabe qué hizo el día anterior y no necesita ponerse en contexto y cuando acaba una entrega también conoce perfectamente qué ha hecho. Todo está en su cabeza y en sus notas, no necesita más.

### 2.5.4 miSEL Software Development Process (miSEL-sdp)

miSEL-sdp es una metodología ágil desarrollada por el grupo de investigación miSEL del *Software Engineering Lab* de la Universidad Carlos III de Madrid. Es una metodología ágil especialmente pensada para el desarrollo de aplicaciones móviles que consta de un ciclo de vida con nueve fases. [17]

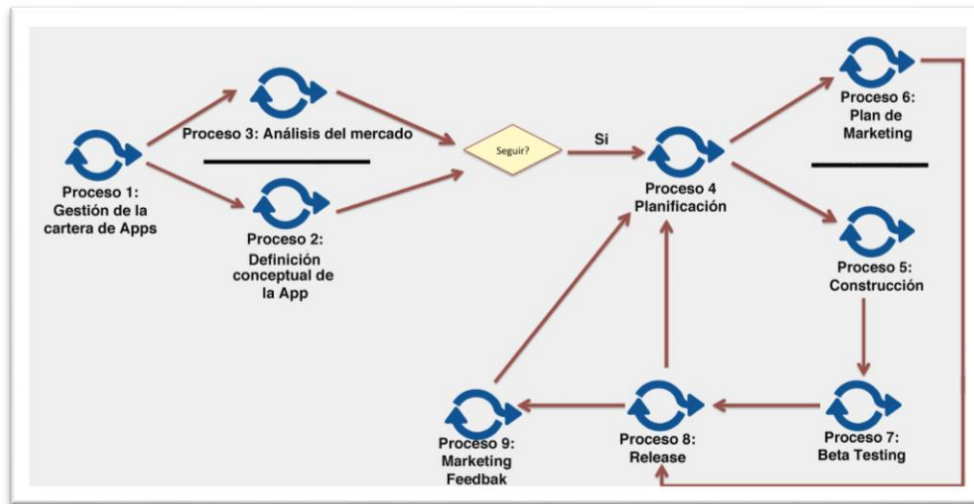


Ilustración 13: Ciclo de vida de miSEL-sdp [17]

En el primer paso –*Gestión de la cartera de Apps*–, se debe pensar en todas las *apps* posibles a desarrollar en función a las necesidades que se presenten; Estas aplicaciones se priorizan y una vez priorizadas, la que tenga la prioridad más alta seleccionada, será la que sea desarrollada.

Después en paralelo se realizan los procesos dos y tres, *Análisis del mercado* y *Definición conceptual de la app*, respectivamente en los cuales se estudia la viabilidad en el mercado de la *app* a desarrollar y se define el producto a desarrollar. Llegados a este punto y con el resultado de estos dos procesos debemos plantearnos si seguir o no. Si se ha llegado a la conclusión de que hay que seguir adelante, se pasa al proceso número cuatro –*Planificación*–.

Durante la fase de planificación, se asignan recursos al proyecto, como personal, tiempo de desarrollo, materiales, etc. En esta fase también se planifican funcionalidades y restricciones.

Los pasos cinco y seis, vuelven a realizarse en paralelo. El proceso cinco –*Construcción*– es el proceso de programación de la aplicación, a la par que se desarrolla la *app* el proceso seis –*Plan de Márquetin*– crea la estrategia comercial de la aplicación. En el proceso siete –*Beta Testing*– se realizan las pruebas internas pre-lanzamiento de la aplicación.

Cuando los *stakeholders* quedan satisfechos con el resultado del proceso siete, se pasa al proceso ocho, –*Release*– en el cual se pone la aplicación a disposición del público y se promociona en el mercado según el plan de márquetin resultante del sexto proceso.

En el noveno proceso –*Marketing feedback*– se estudian las valoraciones y reseñas de los usuarios, realimentando el proceso de planificación y siguiendo todo el ciclo nuevamente para el lanzamiento de la siguiente versión de la *app*.

---

## Capítulo 3: Análisis de la solución

---

---

## 3.1 Introducción

En este capítulo se incluye una descripción del sistema que se ha desarrollado junto con las historias de usuario y una especificación de requisitos software. Para la elaboración de este capítulo se ha seguido parcialmente el estándar IEEE 830-1998 [18].

Las historias de usuario se definen para conocer rápidamente lo que el usuario quiere para poder completar las metas que quiere alcanzar a través del software. No están contempladas en el estándar IEEE 830 pero sí en la metodología de desarrollo escogida, Scrumban.

Finalmente, y conociendo qué desea el usuario, se realiza la especificación formal de requisitos funcionales y no funcionales del sistema.

---

## 3.2 Descripción general del sistema a desarrollar

Siguiendo el estándar IEEE 830, en esta sección se procede a describir los factores que afectan a la solución y a sus requisitos. Se busca definir el contexto en el que se enclavarán los requisitos para poder definir los requisitos con más detalle en secciones posteriores facilitando su comprensión.

### 3.2.1 Perspectiva del producto

Este producto se enmarca como ya se ha dicho en el capítulo *Estado del arte* dentro de un conjunto de aplicaciones de *Mobile Classroom Environment* en desarrollo por el grupo de investigación miSEL, perteneciente al *Software Engineering Lab* de la Universidad Carlos III de Madrid. Dichas aplicaciones buscan proporcionar un entorno de aprendizaje personal completo para uso de alumnos y profesores.

Dicho entorno consta de una serie de aplicaciones que realizan las siguientes funciones para los profesores...

- Planificar y preparar clases
- Impartir clases
- Evaluar a los alumnos

Y para los alumnos

- Seguir el curso de una clase
- Realizar ejercicios
- Realizar exámenes tipo test

### 3.2.2 Funciones del producto

El producto que se pide debe ser cubrir dos grandes bloques de funcionalidad de los que se han enumerado en el apartado *Perspectiva del producto*. Dichos conjuntos de funcionalidad son el conjunto de las funcionalidades para planificar y preparar clases y de la realización de exámenes de tipo test.

Por tanto, el producto debe permitir la planificación de clases y la preparación de todos los contenidos y recursos necesarios para su posterior impartición.

También deben desarrollarse las capacidades de realización de exámenes de tipo test, para ello se deberán poder crear las preguntas y proporcionar toda la infraestructura necesaria para que los alumnos puedan recibirlas en sus dispositivos y responderlas.

### 3.2.3 Características de los usuarios

El producto va dirigido principalmente a usuarios de la comunidad universitaria, tanto alumnos como profesores y que dispongan de un dispositivo con sistema operativo iOS y con conocimientos elementales sobre su uso, tales como instalar aplicaciones o interacciones básicas como navegar por las distintas pantallas de una aplicación, pulsar botones, introducir texto, etc.

Como ya se ha visto, los usuarios se dividen en dos grupos. A saber.

- Profesores: Necesitan planificar por adelantado los contenidos de sus asignaturas y sesiones de clase para luego poder impartirlas con todos los recursos a mano, también buscan mantener la interactividad en clase y lanzar preguntas al alumnado que respondan todos los alumnos y no únicamente los “sospechosos habituales” para mantener el ritmo de la clase y que la audiencia no se distraiga.
- Alumnos: Buscan poder seguir las clases de modo más dinámico y no aburrirse durante las sesiones y perder el foco de atención en lo importante.

### 3.2.4 Restricciones

El producto será desarrollado para la plataforma iOS, con lo cual se tienen las siguientes restricciones.

- El producto debe ser desarrollado empleando el kit de desarrollo proporcionado por Apple y dicho kit solo funciona en ordenadores con el sistema operativo Mac OS X.
- El sistema operativo Mac OS X está soportado únicamente por ordenadores de la marca Apple.
- El lenguaje de programación a emplear debe ser Objective-C, único lenguaje soportado oficialmente por el kit de desarrollo.
- El producto debe seguir la guía de estilo para interfaces de usuario publicadas por Apple [19].
- Se debe usar la red social Twitter para realizar la identificación de los alumnos.
- Se deben usar notificaciones *push* para transmitir las preguntas a los alumnos.
- Las notificaciones *push* para iOS sólo pueden transportar 256 bytes de información adicional.

### 3.2.5 Suposiciones y dependencias

- Los alumnos deben tener un dispositivo iOS propio.
- Los profesores deben tener a su disposición también un dispositivo con iOS.
- Los alumnos deben tener una cuenta twitter para responder a preguntas.
- Cada asignatura debe tener una cuenta de twitter propia.

### 3.3 Alcance de la solución

Parte del análisis de la solución consiste en determinar el alcance de la misma definiendo unos límites que permitan discernir qué es realmente relevante para la resolución del problema que nos ocupa y así focalizar el esfuerzo en aquello que es importante evitando desperdiciar recursos.

La solución debe funcionar en dispositivos móviles con sistema operativo iOS tanto teléfonos como tabletas para abarcar la mayor cantidad de público posible, sobre todo de cara a los alumnos que son los que a priori dispondrán de una gama más variada de dispositivos.

La solución debe permitir a los profesores preparar sus clases con antelación estén donde estén, con un dispositivo e impartirla más tarde ante sus alumnos empleando para ello el mismo dispositivo o incluso otro distinto sin preocupación alguna, plantear cuestiones a los alumnos y analizar estadísticas de las respuestas de los alumnos.

Los alumnos han de recibir en sus dispositivos las preguntas que les planteen sus profesores y tener la capacidad de responderlas. Los alumnos deben quedar identificados a través de nombres de usuario de la red social Twitter.

### 3.4 Historias de usuario

En esta sección se detallarán las historias de usuario que conforman una especificación ágil —a grandes rasgos— de las funcionalidades de la solución, en posteriores secciones del documento estas historias se ampliarán con más detalle y se transformarán en una especificación formal de requisitos.

Antes de empezar a relatar las historias de usuario es conveniente conocer qué es una historia de usuario y qué formato se va a emplear en este documento para su especificación.

Según Mike Cohn [20] las historias de usuario son una simple y breve descripción desde la perspectiva de una persona que desea una nueva funcionalidad en una aplicación y cuya descripción suele presentar una estructura con siguiente aspecto con ligeras variaciones.

*Como <tipo de usuario>, quiero <meta a alcanzar> para <razón>*

Un ejemplo de historia de usuario podría ser este

*Como usuario, quiero poder copiar texto para exportarlo a otro programa*

Las historias pueden abarcar mayor o menor cantidad de funcionalidad, las que abarcan una gran cantidad de funcionalidad y no son abarcables, tienen que ser subdivididas en historias más simples. Este tipo de grandes historias se llaman —muy apropiadamente— historias épicas.

El formato que se va a emplear en la definición del *backlog* de historias de usuario es el que se propone en [21] donde Mike Cohn explica las ventajas de emplear este formato.

Identificador	Como...	Quiero...	Para...
US-001	Usuario	Poder copiar texto	Exportarlo a otro programa

Tabla 2: Ejemplo de backlog con historias de usuario



### 3.4.1 Product backlog

Empleando el formato descrito anteriormente se procede en esta sección a mostrar el *product backlog* del proyecto.

Identificador	Como...	Quiero...	Para...
<b>US-001</b>	Profesor	Poder gestionar los cursos que imparto	Organizar los contenidos de mis cursos.
<b>US-002</b>	Profesor	Poder gestionar las sesiones de clase de cada curso	Estructurar los contenidos de cada uno de mis cursos en sesiones.
<b>US-003</b>	Profesor	Poder gestionar los conceptos que voy a impartir	Estructurar los contenidos de cada una de mis sesiones en bloques temáticos conceptuales.
<b>US-004</b>	Profesor	Poder gestionar las preguntas que deseo realizar en clase	Enviar preguntas a mis alumnos durante la clase y evaluar sus progresos al momento.
<b>US-005</b>	Profesor	Poder preparar tweets y vincularlos a una sesión	Enviar dichos tweets durante una sesión de clase y que los lea mi alumnado.
<b>US-006</b>	Profesor	Poder preparar mis clases allá donde vaya	Incorporar nuevas ideas y contenidos en cualquier momento y lugar.
<b>US-007</b>	Profesor	Poder ver estadísticas de las respuestas de mis alumnos	Evaluar a mis alumnos.
<b>US-008</b>	Alumno	Identificarme con mi cuenta de Twitter	Aprovechar la cuenta que utilizo para preguntar dudas en clase y no tener una cuenta distinta para cada actividad de clase.
<b>US-009</b>	Alumno	Suscribirme a los cursos que me interesen	Recibir preguntas y participar en los cursos que estoy matriculado.
<b>US-010</b>	Alumno	Poder recibir preguntas sin preocuparme todo el rato por mi dispositivo	Atender a la clase y no sufrir distracciones innecesarias.
<b>US-011</b>	Alumno	Responder las preguntas que me formulen	Participar en las evaluaciones que mi profesor haga.
<b>US-012</b>	Alumno	Tener una lista de las preguntas recibidas y poder ver cuales he leído y cuales son nuevas	Recibir múltiples preguntas y saber cuáles he respondido ya y cuáles no.

Tabla 3: Product backlog

## 3.5 Especificación de requisitos

En el siguiente catálogo se describen todos los requisitos que forman parte del proyecto para conformar una especificación completa a partir de la cual diseñar la solución al problema planteado al inicio de este documento.

### 3.5.1 Formato de los requisitos

A continuación se describen los campos que definen cada uno de los requisitos.

- **Identificador:** Código que identifica de manera única al requisito. Tendrá el siguiente formato, **REQ-XX** dónde **REQ** representa el tipo de requisito que se define.
- **Nombre:** Descripción de la función del requisito de forma abreviada.
- **Descripción:** Funcionalidad del requisito, expresado de manera completa, sin ambigüedades ni contradicciones.
- **Historia de usuario:** Código identificador de la historia de usuario de la que proviene este requisito.

Parámetro	Descripción
<b>Identificador</b>	REQ-XXX
<b>Nombre</b>	Nombre del requisito
<b>Descripción</b>	Descripción del requisito
<b>Prioridad</b>	Alta – Media – Baja
<b>Estabilidad</b>	Estable – Inestable
<b>Necesidad</b>	Esencial – Deseable – Opcional
<b>Verificabilidad</b>	Alta – Media – Baja – N/A
<b>Historia de usuario</b>	US-XXX

Tabla 4: Plantilla para la definición de requisitos

### 3.5.2 Catálogo de requisitos

A continuación se exponen todos los requisitos a los que la solución debe responder para ser válida. El catálogo se subdivide en requisitos funcionales —que especifican los servicios y funcionalidades de un sistema— y en requisitos no funcionales, que especifican las restricciones que deben existir en el producto ya desarrollado, no durante el proceso de desarrollo.

#### 3.5.2.1 Requisitos funcionales

En este apartado se describen todas y cada una de las funcionalidades que debe poseer la solución

Parámetro	Descripción
<b>Identificador</b>	RF-001
<b>Nombre</b>	Crear curso
<b>Descripción</b>	Los profesores podrán dar de alta nuevos cursos. Un nuevo curso se creará a partir de los siguientes datos: <ul style="list-style-type: none"> <li>• Nombre del curso.</li> <li>• Descripción del curso.</li> <li>• Cuenta de Twitter que se empleará durante el curso.</li> </ul>
<b>Historia de usuario</b>	US-001

Tabla 5: Requisito RF-001 — Crear curso

Parámetro	Descripción
<b>Identificador</b>	RF-002
<b>Nombre</b>	Modificar curso
<b>Descripción</b>	Los profesores podrán modificar cursos existentes. De un curso pueden modificarse los siguientes datos: <ul style="list-style-type: none"> <li>• Nombre del curso.</li> <li>• Descripción del curso.</li> <li>• Cuenta de Twitter que se empleará durante el curso.</li> </ul>
<b>Historia de usuario</b>	US-001

Tabla 6: Requisito RF-002 — Modificar curso

Parámetro	Descripción
<b>Identificador</b>	RF-003
<b>Nombre</b>	Borrar curso
<b>Descripción</b>	Los profesores podrán borrar cursos existentes
<b>Historia de usuario</b>	US-001

Tabla 7: Requisito RF-003 — Borrar curso

Parámetro	Descripción
<b>Identificador</b>	RF-004
<b>Nombre</b>	Crear sesión
<b>Descripción</b>	Los profesores podrán dar de alta nuevas sesiones. Una nueva sesión se creará a partir de los siguientes datos: <ul style="list-style-type: none"> <li>• Título de la sesión.</li> <li>• Fecha de la sesión.</li> <li>• Descripción de la sesión.</li> <li>• Curso al que está asignada la sesión, por defecto una sesión puede crearse y no asignarse a un curso para hacerlo posteriormente.</li> </ul>
<b>Historia de usuario</b>	US-002

Tabla 8: Requisito RF-004 — Crear sesión

Parámetro	Descripción
<b>Identificador</b>	RF-005
<b>Nombre</b>	Modificar sesión
<b>Descripción</b>	Los profesores podrán modificar sesiones. De una sesión se pueden modificar los siguientes datos: <ul style="list-style-type: none"> <li>• Título de la sesión.</li> <li>• Fecha de la sesión.</li> <li>• Descripción de la sesión.</li> <li>• Curso al que está asignada la sesión, por defecto una sesión puede crearse y no asignarse a un curso para hacerlo posteriormente.</li> </ul>
<b>Historia de usuario</b>	US-002

Tabla 9: Requisito RF-005 — Modificar sesión

Parámetro	Descripción
<b>Identificador</b>	RF-006
<b>Nombre</b>	Borrar sesión
<b>Descripción</b>	Los profesores podrán borrar sesiones existentes
<b>Historia de usuario</b>	US-002

Tabla 10: Requisito RF-006 — Borrar sesión

Parámetro	Descripción
<b>Identificador</b>	RF-007
<b>Nombre</b>	Crear concepto
<b>Descripción</b>	Los profesores podrán dar de alta nuevos conceptos. Un nuevo concepto se creará a partir de los siguientes datos: <ul style="list-style-type: none"> <li>• Título del concepto.</li> <li>• Descripción del concepto.</li> <li>• Curso al que está asignado el concepto, por defecto un concepto puede crearse y no asignarse a un curso para hacerlo posteriormente.</li> <li>• Sesión a la que está asignada el concepto, por defecto un concepto puede crearse y no asignarse a una sesión para hacerlo posteriormente.</li> </ul>
<b>Historia de usuario</b>	US-003

Tabla 11: Requisito RF-007 — Crear concepto

Parámetro	Descripción
<b>Identificador</b>	RF-008
<b>Nombre</b>	Modificar concepto
<b>Descripción</b>	Los profesores podrán modificar conceptos. De un concepto pueden modificarse los siguientes datos: <ul style="list-style-type: none"> <li>• Título del concepto.</li> <li>• Descripción del concepto.</li> <li>• Curso al que está asignado el concepto, por defecto un concepto puede crearse y no asignarse a un curso para hacerlo posteriormente.</li> <li>• Sesión a la que está asignada el concepto, por defecto un concepto puede crearse y no asignarse a una sesión para hacerlo posteriormente.</li> </ul>
<b>Historia de usuario</b>	US-003

Tabla 12: Requisito RF-008 — Modificar concepto

Parámetro	Descripción
<b>Identificador</b>	RF-009
<b>Nombre</b>	Borrar curso
<b>Descripción</b>	Los profesores podrán borrar cursos existentes
<b>Historia de usuario</b>	US-003

Tabla 13: Requisito RF-009 — Borrar concepto

Parámetro	Descripción
<b>Identificador</b>	RF-010
<b>Nombre</b>	Crear pregunta
<b>Descripción</b>	Los profesores podrán dar de alta nuevas preguntas. Un nuevo concepto se creará a partir de los siguientes datos: <ul style="list-style-type: none"> <li>• Texto de la pregunta.</li> <li>• Descripción del concepto.</li> <li>• Concepto al que está asignada la pregunta, por defecto una pregunta puede crearse y no asignarse a un concepto para hacerlo posteriormente.</li> <li>• De una a n respuestas</li> </ul>
<b>Historia de usuario</b>	US-004

Tabla 14: Requisito RF-010 — Crear pregunta

Parámetro	Descripción
<b>Identificador</b>	RF-011
<b>Nombre</b>	Modificar pregunta
<b>Descripción</b>	Los profesores podrán modificar preguntas. De una pregunta pueden modificarse los siguientes datos: <ul style="list-style-type: none"> <li>• Texto de la pregunta.</li> <li>• Descripción del concepto.</li> <li>• Concepto al que está asignada la pregunta, por defecto una pregunta puede crearse y no asignarse a un concepto para hacerlo posteriormente.</li> <li>• Respuestas (sólo pueden borrarse y añadir nuevas)</li> </ul>
<b>Historia de usuario</b>	US-004

Tabla 15: Requisito RF-011 — Modificar pregunta

Parámetro	Descripción
<b>Identificador</b>	RF-012
<b>Nombre</b>	Borrar pregunta
<b>Descripción</b>	Los profesores podrán borrar preguntas existentes
<b>Historia de usuario</b>	US-004

Tabla 16: Requisito RF-012 — Borrar pregunta

Parámetro	Descripción
<b>Identificador</b>	RF-013
<b>Nombre</b>	Crear tweet
<b>Descripción</b>	Los profesores podrán crear tweets en la aplicación, pero no publicarlos.
<b>Historia de usuario</b>	US-005

Tabla 17: Requisito RF-013 — Crear tweet

Parámetro	Descripción
<b>Identificador</b>	RF-014
<b>Nombre</b>	Borrar tweet
<b>Descripción</b>	Los profesores podrán borrar tweets que hayan preparado con anterioridad, pero si han sido publicados la acción no los borrará de Twitter.
<b>Historia de usuario</b>	US-005

Tabla 18: Requisito RF-014 — Borrar tweet

Parámetro	Descripción
<b>Identificador</b>	RF-015
<b>Nombre</b>	Consultar estadísticas
<b>Descripción</b>	Los profesores deben poder ser capaces de consultar las estadísticas de respuestas de sus alumnos por sesión y pregunta (porcentaje de aciertos) y un listado de los alumnos que han respondido cada opción.
<b>Historia de usuario</b>	US-007

Tabla 19: Requisito RF-015 — Consultar estadísticas

Parámetro	Descripción
<b>Identificador</b>	RF-016
<b>Nombre</b>	Sincronización <i>cloud</i>
<b>Descripción</b>	La aplicación debe ser capaz de sincronizarse con un servidor para compartir información entre distintos dispositivos. Dicha información también debe poder estar a disposición de otras aplicaciones para el profesor que se deseen integrar en la solución.
<b>Historia de usuario</b>	US-006

Tabla 20: Requisito RF-016 — Sincronización *cloud*

Parámetro	Descripción
<b>Identificador</b>	RF-017
<b>Nombre</b>	Identificación con twitter
<b>Descripción</b>	Los alumnos deben ser capaces de identificarse en el sistema usando una cuenta de usuario de Twitter.
<b>Historia de usuario</b>	US-008

Tabla 21: Requisito RF-017 — Identificación con Twitter

Parámetro	Descripción
<b>Identificador</b>	RF-018
<b>Nombre</b>	Suscribirse a una sesión
<b>Descripción</b>	Los alumnos deben poder suscribirse a una o varias sesiones abiertas de todas las que haya disponibles para recibir las preguntas de dichas sesiones.
<b>Historia de usuario</b>	US-009

Tabla 22: Requisito RF-018 — Suscribirse a una sesión

Parámetro	Descripción
<b>Identificador</b>	RF-019
<b>Nombre</b>	Recibir preguntas
<b>Descripción</b>	<p>Los alumnos deben poder recibir en sus dispositivos las preguntas que lancen los profesores de las sesiones a las que están suscritos.</p> <p>Si tienen la aplicación abierta, cuando se lance la pregunta, esta deberá colocarse inmediatamente en pantalla para su respuesta.</p> <p>En caso contrario, el dispositivo del alumno mostrará una notificación que al interactuar con ella le lleve directamente a la pregunta.</p>
<b>Historia de usuario</b>	US-010

Tabla 23: Requisito RF-019 — Recibir preguntas

Parámetro	Descripción
<b>Identificador</b>	RF-020
<b>Nombre</b>	Responder preguntas
<b>Descripción</b>	Los alumnos deben poder responder a las preguntas que reciban en sus dispositivos mediante el propio dispositivo, seleccionando la opción que consideren correcta y poder modificar su respuesta durante todo el tiempo que la sesión a la que la pregunta pertenece esté abierta.
<b>Historia de usuario</b>	US-011

Tabla 24: Requisito RF-020 — Responder preguntas

Parámetro	Descripción
<b>Identificador</b>	RF-021
<b>Nombre</b>	Listar preguntas recibidas
<b>Descripción</b>	Los alumnos deben poder ver las preguntas que han recibido en una lista de tipo “bandeja de entrada” donde puedan distinguir qué preguntas han visitado ya y cuales son nuevas. Deben asimismo tener la posibilidad de seleccionar la que crean oportuna y así acceder a la pregunta propiamente dicha.
<b>Historia de usuario</b>	US-012

Tabla 25: Requisito RF-021 — Listar preguntas recibidas

### 3.5.2.2 Requisitos no funcionales

En esta sección se describen los requisitos que debe cumplir la solución pero que no son funcionalidades si no aspectos a los que el producto final debe ajustarse.

Parámetro	Descripción
<b>Identificador</b>	RNF-001
<b>Nombre</b>	Dispositivos soportados
<b>Descripción</b>	La solución debe funcionar en los dispositivos que funcionen con el sistema operativo iOS, ya sean teléfonos móviles o tabletas.
<b>Prioridad</b>	Alta
<b>Estabilidad</b>	Estable
<b>Necesidad</b>	Esencial
<b>Verificabilidad</b>	Alta
<b>Historia de usuario</b>	N/A

Tabla 26: Requisito RNF-001 — Dispositivos soportados

Parámetro	Descripción
<b>Identificador</b>	RNF-002
<b>Nombre</b>	Versión mínima del sistema operativo iOS
<b>Descripción</b>	La versión mínima del sistema iOS sobre la que funcionará la solución es la versión 6.0
<b>Historia de usuario</b>	N/A

Tabla 27: Requisito RNF-002 — Versión mínima del sistema operativo iOS

Parámetro	Descripción
<b>Identificador</b>	RNF-003
<b>Nombre</b>	Interfaz adaptativa
<b>Descripción</b>	La interfaz de usuario debe adaptarse a los distintos tamaños de pantalla de los teléfonos iPhone y tabletas iPad.
<b>Historia de usuario</b>	N/A

Tabla 28: Requisito RNF-003 — Interfaz adaptativa

Parámetro	Descripción
<b>Identificador</b>	RNF-004
<b>Nombre</b>	Uso del patrón de interfaz <i>Pull to refresh</i>
<b>Descripción</b>	Aquellas partes de la interfaz de usuario en las que pueda solicitarse manualmente una actualización de la misma deberá solicitarse empleando el patrón <i>Pull to refresh</i> .
<b>Historia de usuario</b>	N/A

Tabla 29: Requisito RNF-004 — Uso del patrón de interfaz *Pull to refresh*

Parámetro	Descripción
<b>Identificador</b>	RNF-005
<b>Nombre</b>	Seguimiento de las <i>Human Interface Guidelines</i> de Apple
<b>Descripción</b>	La interfaz de usuario debe ser diseñada conforme a las directrices especificadas en [19]
<b>Historia de usuario</b>	N/A

Tabla 30: Requisito RNF-005 — Seguimiento de las *Human Interface Guidelines* de Apple



### 3.5.3 Trazabilidad de requisitos – historias de usuario

En esta sección se muestra la correspondencia entre historias de usuario y requisitos.

	US-001	US-002	US-003	US-004	US-005	US-006	US-007	US-008	US-009	US-010	US-011	US-012
RF-001	X											
RF-002	X											
RF-003	X											
RF-004		X										
RF-005		X										
RF-006		X										
RF-007			X									
RF-008			X									
RF-009			X									
RF-010				X								
RF-011				X								
RF-012				X								
RF-013					X							
RF-014					X							
RF-015							X					
RF-016						X						
RF-017								X				
RF-018									X			
RF-019										X		
RF-020											X	
RF-021												X

Tabla 31: Matriz de trazabilidad requisitos - historias de usuario

## 3.6 Especificación de casos de uso

En la presente sección del documento se elaborará una especificación de los casos de uso del sistema. Primero se presentará el diagrama de casos de uso en lenguaje gráfico UML y a continuación se procederá a realizar una descripción textual pormenorizada de cada caso de uso.

Para la descripción textual se empleará una tabla con los siguientes campos:

- **Identificador:** Identifica a cada caso de uso. El formato utilizado es el siguiente “**CU-XX**”, donde cada “**X**” se corresponde con un número comprendido entre 0 y 9.
- **Nombre:** Se corresponde con el nombre asignado al caso de uso.
- **Actores:** Son los distintos roles de usuarios que encontraremos en el sistema. Es cualquier entidad externa al Sistema que está relacionada con éste y que le demanda una funcionalidad.
- **Objetivo:** Se corresponde con la finalidad que se pretende obtener tras realizarse el caso de uso.
- **Curso típico de eventos:** Se corresponde con la descripción del caso de uso referido.
- **Pre-condición:** Se corresponde con las condiciones iniciales que se deben cumplir para que se realice dicho caso de uso.
- **Post-condición:** Se corresponde con las condiciones finales que se cumplirán tras realizar dicho caso de uso.

### 3.6.1 Diagrama de casos de uso

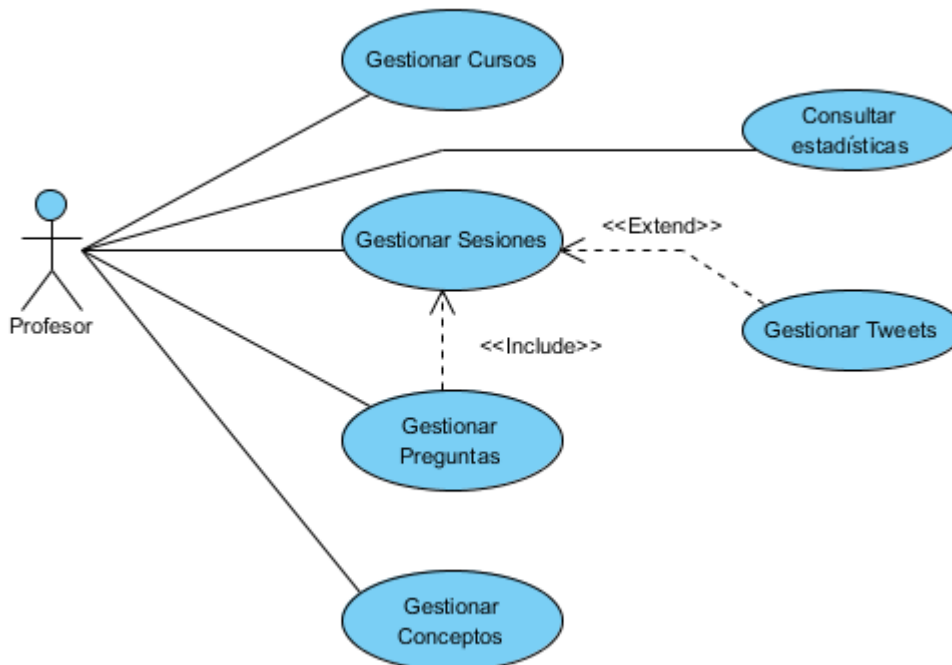


Ilustración 14: Diagrama de casos de uso para el actor Profesor

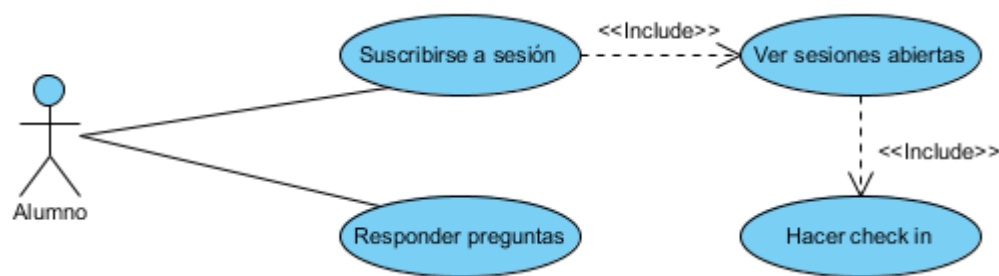


Ilustración 15: Diagrama de casos de uso para el actor Alumno

3.6.2 Descripción textual

A continuación se procede a realizar la descripción textual de los casos de uso.

Identificador	CU-001
Nombre	Gestionar cursos
Actores	Profesor
Objetivo	El profesor desea crear, modificar, consultar o borrar cursos del sistema.
Curso típico de eventos	<ol style="list-style-type: none"><li>1. El profesor abre la aplicación.</li><li>2. El profesor selecciona la opción ‘Cursos’ en el menú principal.</li><li>3. El profesor se encuentra con la pantalla de gestión de cursos.</li><li>4. El profesor realiza la acción deseada.</li></ol>
Pre-condición	<ul style="list-style-type: none"><li>• La aplicación debe encontrarse iniciada.</li><li>• El menú principal debe estar visible.</li></ul>
Post-condición	El sistema registra los cambios que se hayan producido.

Tabla 32: Caso de uso CU-001 — Gestionar cursos

Identificador	CU-002
Nombre	Gestionar sesiones
Actores	Profesor
Objetivo	El profesor desea crear, modificar, consultar o borrar sesiones del sistema.
Curso típico de eventos	<ol style="list-style-type: none"><li>1. El profesor abre la aplicación.</li><li>2. El profesor selecciona la opción ‘Sesiones’ en el menú principal.</li><li>3. El profesor se encuentra con la pantalla de gestión de sesiones.</li><li>4. El profesor realiza la acción deseada.</li></ol>
Pre-condición	<ul style="list-style-type: none"><li>• La aplicación debe encontrarse iniciada.</li><li>• El menú principal debe estar visible.</li></ul>
Post-condición	El sistema registra los cambios que se hayan producido.

Tabla 33: Caso de uso CU-002 — Gestionar sesiones

Identificador	CU-003
Nombre	Gestionar tweets
Actores	Profesor
Objetivo	El profesor desea crear, consultar o borrar tweets del sistema.
Curso típico de eventos	<ol style="list-style-type: none"><li>1. El profesor abre la aplicación.</li><li>2. El profesor inicia la gestión de una sesión [CU-002]</li><li>3. El profesor selecciona la opción tweets y se encuentra con la pantalla de gestión de tweets.</li><li>4. El profesor realiza la acción deseada.</li></ol>
Pre-condición	<ul style="list-style-type: none"><li>• La aplicación debe encontrarse iniciada.</li><li>• El texto a introducir no puede tener más de 140 caracteres</li></ul>
Post-condición	El sistema registra los cambios que se hayan producido.

Tabla 34: Caso de uso CU-003 — Gestionar tweets

Identificador	CU-004
Nombre	Gestionar preguntas
Actores	Profesor
Objetivo	El profesor desea crear, modificar, consultar o borrar preguntas del sistema.
Curso típico de eventos	<ol style="list-style-type: none"> <li>1. El profesor abre la aplicación.</li> <li>2. El profesor selecciona la opción 'Preguntas' en el menú principal.</li> <li>3. El profesor se encuentra con la pantalla de gestión de preguntas.</li> <li>4. El profesor realiza la acción deseada.</li> </ol>
Pre-condición	<ul style="list-style-type: none"> <li>• La aplicación debe encontrarse iniciada.</li> <li>• El menú principal debe estar visible.</li> </ul>
Post-condición	El sistema registra los cambios que se hayan producido.

Tabla 35: Caso de uso CU-004 — Gestionar preguntas

Identificador	CU-005
Nombre	Gestionar conceptos
Actores	Profesor
Objetivo	El profesor desea crear, modificar, consultar o borrar conceptos del sistema.
Curso típico de eventos	<ol style="list-style-type: none"> <li>1. El profesor abre la aplicación.</li> <li>2. El profesor selecciona la opción 'Conceptos' en el menú principal.</li> <li>3. El profesor se encuentra con la pantalla de gestión de conceptos.</li> <li>4. El profesor realiza la acción deseada.</li> </ol>
Pre-condición	<ul style="list-style-type: none"> <li>• La aplicación debe encontrarse iniciada.</li> <li>• El menú principal debe estar visible.</li> </ul>
Post-condición	El sistema registra los cambios que se hayan producido.

Tabla 36: Caso de uso CU-005 — Gestionar conceptos

Identificador	CU-006
Nombre	Hacer <i>check in</i>
Actores	Alumno
Objetivo	El alumno desea iniciar sesión en el sistema para poder atender a las clases en las que está interesado
Curso típico de eventos	<ol style="list-style-type: none"> <li>1. El abre la aplicación.</li> <li>2. Al alumno se le muestra una serie de cuentas de twitter disponibles en su dispositivo</li> <li>3. El alumno selecciona la cuenta con la que desea identificarse y pulsa el botón de inicio de sesión.</li> </ol>
Pre-condición	<ul style="list-style-type: none"> <li>• El alumno debe haber abierto la aplicación</li> <li>• El alumno debe haber configurado al menos una cuenta twitter en su dispositivo</li> <li>• Debe existir conexión a Internet</li> </ul>
Post-condición	El alumno queda registrado en el sistema

Tabla 37: Caso de uso CU-006 — Hacer *check in*

<b>Identificador</b>	<b>CU-007</b>
<b>Nombre</b>	Ver sesiones abiertas
<b>Actores</b>	Alumno
<b>Objetivo</b>	El alumno desea ver qué sesiones hay actualmente en curso.
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno hace <i>check in</i> en el sistema [Tabla 37: Caso de uso CU-006 — Hacer <i>check in</i>].</li> <li>2. El alumno recibe y ve una lista de las sesiones que hay actualmente en curso en el sistema.</li> </ol>
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>• El alumno debe haber hecho <i>check in</i> en el sistema.</li> </ul>
<b>Post-condición</b>	El sistema muestra una lista de sesiones en curso.

Tabla 38: Caso de uso CU-007 — Ver sesiones abiertas

<b>Identificador</b>	<b>CU-008</b>
<b>Nombre</b>	Suscribirse a sesión
<b>Actores</b>	Alumno
<b>Objetivo</b>	El alumno desea suscribirse a una o varias sesiones en curso
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno consulta qué sesiones están abiertas [Tabla 39: Caso de uso CU-008 — ].</li> <li>2. El alumno, marca activando una casilla a qué sesiones quiere suscribirse.</li> </ol>
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>• El alumno debe haber hecho <i>check in</i> en el sistema.</li> </ul>
<b>Post-condición</b>	El sistema muestra activas las casillas que el alumno ha marcado si se ha suscrito con éxito, en caso contrario las deja en su estado original.

Tabla 39: Caso de uso CU-008 — Suscribirse a sesión

<b>Identificador</b>	<b>CU-009</b>
<b>Nombre</b>	Responder preguntas
<b>Actores</b>	Alumno
<b>Objetivo</b>	El alumno desea responder a una pregunta recibida
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El alumno recibe una pregunta en su dispositivo</li> <li>2. Se muestra la pregunta, el alumno la lee y elige una de las opciones disponibles de respuesta.</li> <li>3. El alumno pulsa el botón de envío de la respuesta y envía su respuesta.</li> </ol>
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>• El alumno debe haber hecho <i>check in</i> en el sistema.</li> <li>• El alumno debe haberse suscrito a al menos una sesión</li> </ul>
<b>Post-condición</b>	El sistema recibe y procesa la respuesta y marca como leída la pregunta en el dispositivo del alumno.

Tabla 40: Caso de uso CU-009 — Responder preguntas

<b>Identificador</b>	<b>CU-010</b>
<b>Nombre</b>	Consultar estadísticas
<b>Actores</b>	Profesor
<b>Objetivo</b>	El profesor desea consultar estadísticas sobre las preguntas que ha realizado durante una sesión.
<b>Curso típico de eventos</b>	<ol style="list-style-type: none"> <li>1. El profesor abre la aplicación.</li> <li>2. El profesor selecciona la opción 'Estadísticas' en el menú principal.</li> <li>3. El profesor se encuentra con la pantalla de consulta de estadísticas</li> <li>4. El profesor realiza la consulta que desee</li> </ol>
<b>Pre-condición</b>	<ul style="list-style-type: none"> <li>• La aplicación debe encontrarse iniciada.</li> <li>• El menú principal debe estar visible.</li> <li>• Debe existir conexión a Internet</li> </ul>
<b>Post-condición</b>	N/A

Tabla 41: Caso de uso CU-009 — Responder preguntas

### 3.6.3 Trazabilidad requisitos – casos de uso

En esta sección se muestra la correspondencia entre casos de uso y requisitos.

	CU-001	CU -002	CU -003	CU -004	CU -005	CU -006	CU -007	CU -008	CU -009	CU-010
RF-001	X									
RF-002	X									
RF-003	X									
RF-004		X								
RF-005		X								
RF-006		X								
RF-007					X					
RF-008					X					
RF-009					X					
RF-010				X						
RF-011				X						
RF-012				X						
RF-013			X							
RF-014			X							
RF-015										X
RF-016										
RF-017						X				
RF-018								X		
RF-019									X	
RF-020									X	
RF-021							X			

Tabla 42: Matriz de trazabilidad requisitos - casos de uso

---

## Capítulo 4: Diseño de la solución

---

## 4.1 Diseño arquitectónico del sistema

En esta sección se va a tratar el diseño de la arquitectura del sistema exponiendo las alternativas arquitectónicas tanto del sistema en sí como de su entorno operacional.

El problema principal a la hora de vertebrar la arquitectura del sistema es que los dispositivos de los alumnos han de comunicarse con el profesor de algún modo, pero también existe un problema y es que los dispositivos del profesor deben permanecer sincronizados entre sí.

### 4.1.1 Primera alternativa

En la primera alternativa de diseño se planteó realizar dos aplicaciones separadas:

- Una aplicación para que el profesor prepare sus clases y disponga de toda la funcionalidad necesaria para la tarea y también parte de la funcionalidad de impartir la clase, lanzando las preguntas a la audiencia.
- Otra aplicación más ligera destinada únicamente al alumnado para recibir y responder preguntas.

Las aplicaciones se comunicarían vía P2P para enviar las preguntas.

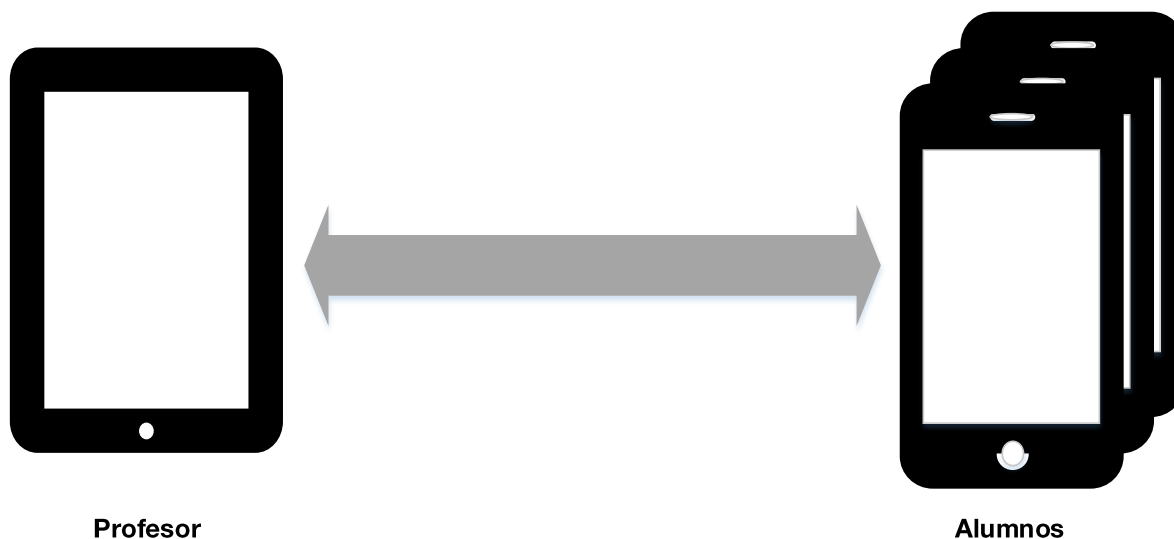


Ilustración 16: Primera alternativa arquitectónica

### 4.1.2 Segunda alternativa

La segunda alternativa comparte elementos con la primera pues también se estructura en dos aplicaciones separadas, una para profesores y otra para alumnos con exactamente el mismo reparto funcionalidades que en la primera alternativa.

En esta segunda alternativa se incluye un tercer elemento, un servicio web intermedio para el envío de preguntas a los alumnos y para la recepción y el procesamiento de las respuestas. El servicio web es también el encargado de alertar a los dispositivos de los alumnos sobre la existencia de nuevas preguntas y de proporcionárselas cuando estos requieran su descarga.

Para proporcionar la ubicuidad de contenidos al profesor, el servicio web será el encargado de almacenar los datos necesarios para mantener sincronizados los dispositivos de los profesores.



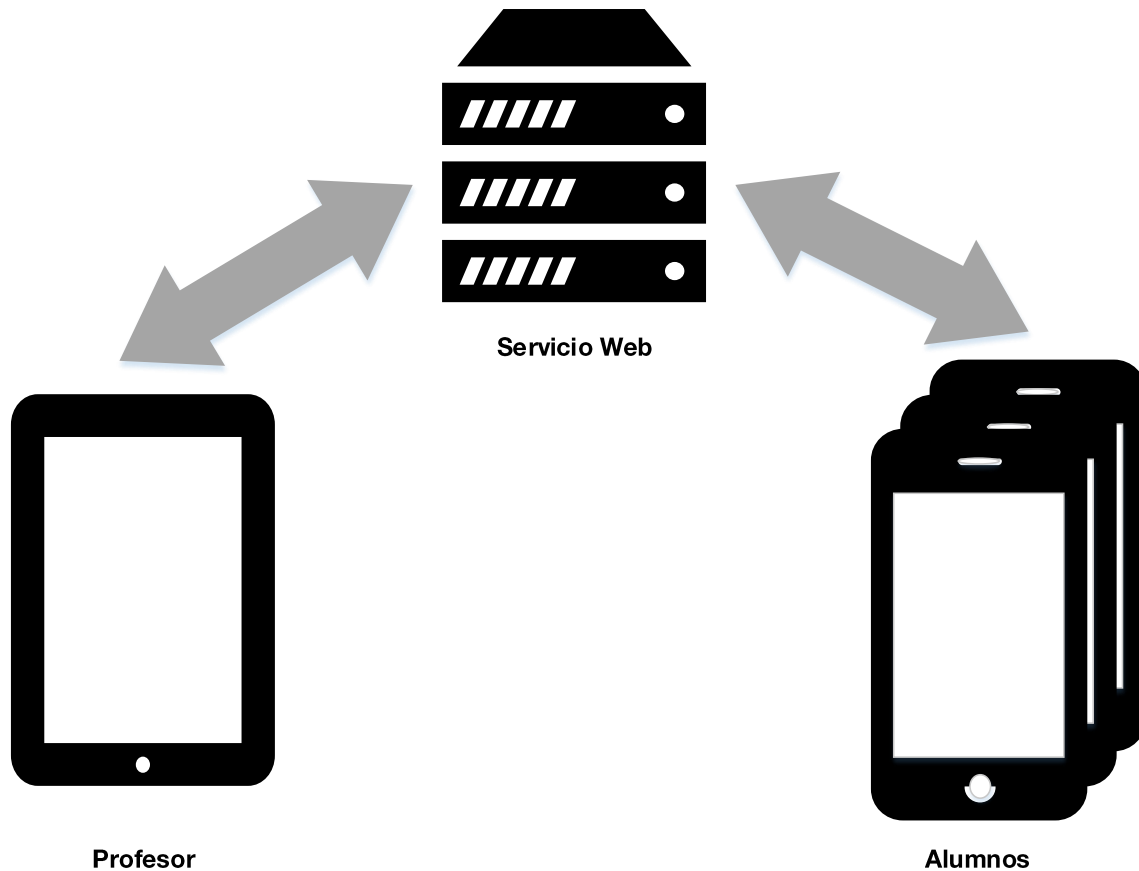


Ilustración 17: Segunda alternativa arquitectónica

#### 4.1.3 Tercera alternativa

La tercera alternativa arquitectónica es una evolución de la segunda que consiste en un desacoplamiento de la funcionalidad de sincronización para aprovechar servicios existentes de sincronización en la nube como iCloud o Dropbox.

La ventaja que supone esta alternativa es que toda la funcionalidad de sincronización y almacenamiento de la información queda proporcionada por dichos servicios evitando tener que reinventar la rueda con un sistema de almacenamiento y sincronización propio, también posee alguna desventaja como la creación de una dependencia o atadura en mayor o menor medida con alguno de los servicios, lo cual podría ser paliado en un hipotético futuro mediante algún proceso migratorio automatizado, mucho menos costoso y más rápido de desarrollar.

Salvo lo mencionado anteriormente, se mantienen todos los demás elementos de la segunda alternativa arquitectónica.

Por ser la alternativa que más flexibilidad para el desarrollo y posibilidades de escalado e interoperabilidad aporta de cara al futuro, esta es la alternativa de diseño elegida para ser desarrollada.

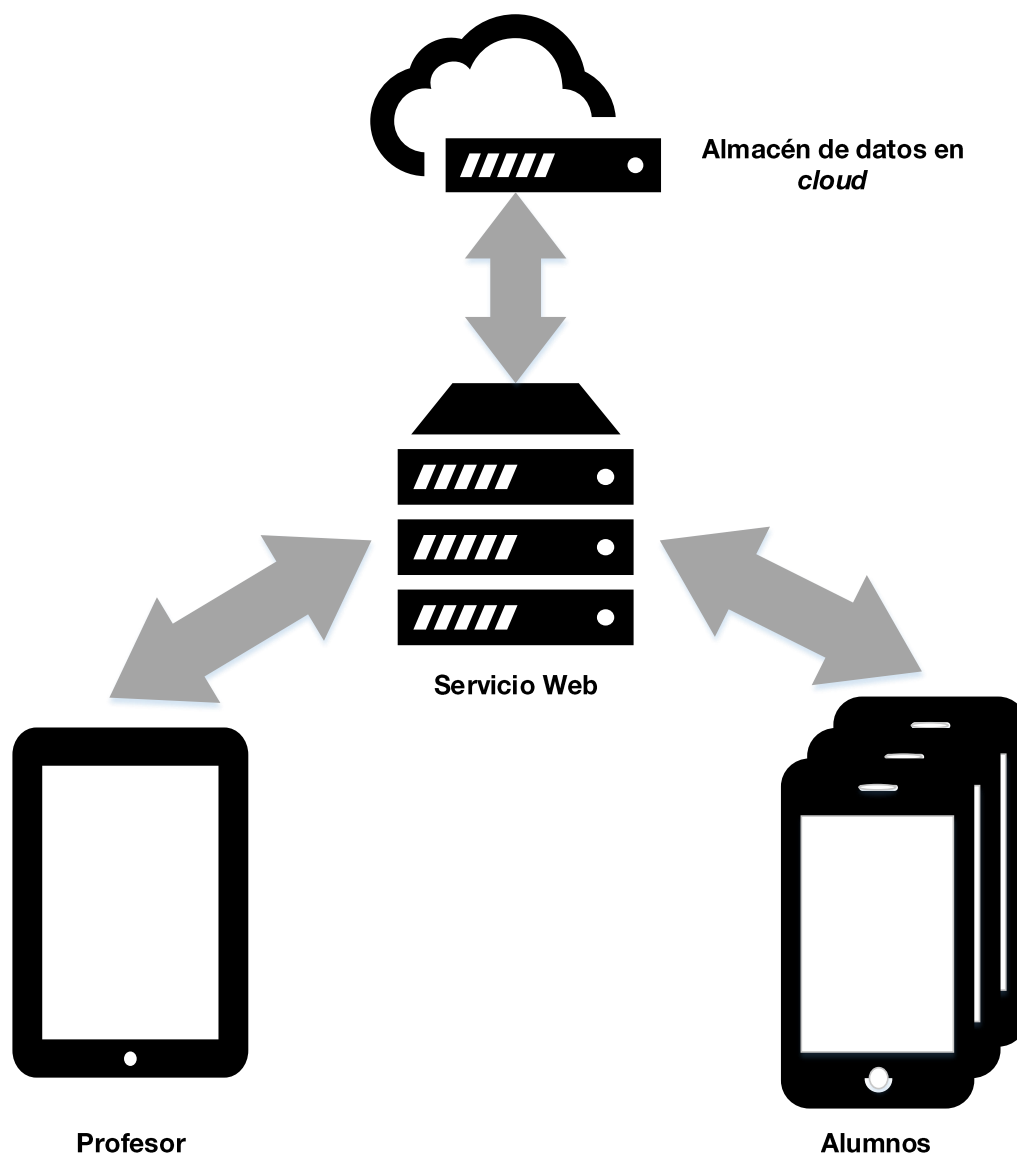


Ilustración 18: Tercera alternativa arquitectónica

#### 4.1.4 Diagrama de componentes

Una vez elegida la alternativa de diseño del sistema a desarrollar, se ha modelado la arquitectura de aplicación utilizando diagramas en lenguaje UML.

El diagrama utilizado en esta sección es el diagrama de componentes. El diagrama de componentes modela la arquitectura software de la solución, agrupando diferentes partes de la aplicación y mostrando la relación proveedor – consumidor de servicios entre componentes.

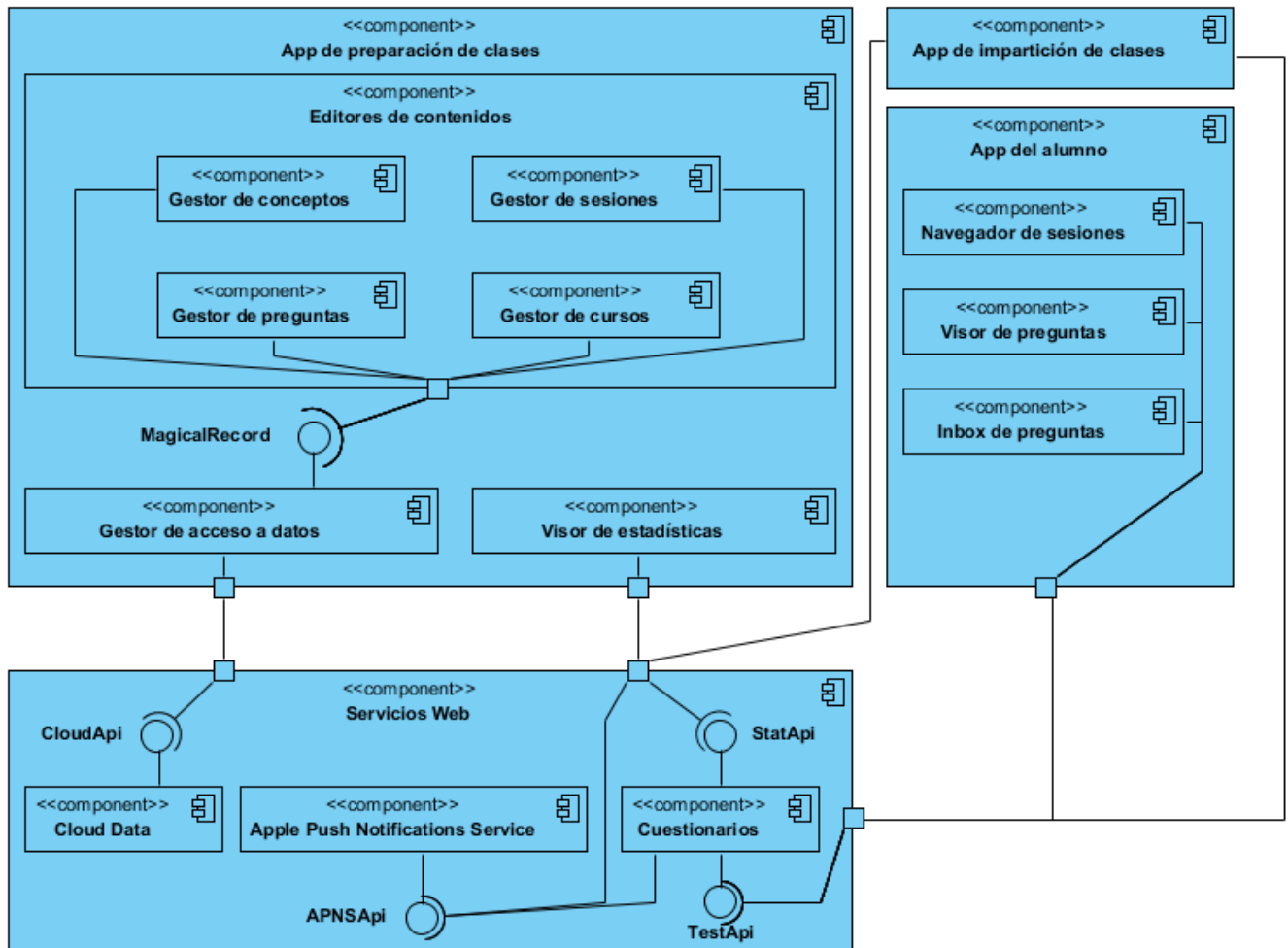


Ilustración 19: Diagrama de componentes

Como puede verse en el diagrama la solución queda estructurada en tres aplicaciones, dos desarrolladas dentro de este proyecto —una para la preparación de clases, otra para que los alumnos sigan la clase—, una tercera aplicación externa para la impartición de las clases desarrollada por José Ángel Cuadrado en [7] y una serie de servicios web que proporcionan los servicios necesarios para interconectar las tres aplicaciones.

## 4.2 Diseño detallado del sistema

En la presente sección se pasará a explicar en líneas generales la función que desempeña cada componente dentro de la solución a través de un enfoque *top-down*, es decir, se irá de lo general a lo particular.

### 4.2.1 Aplicación de preparación de clases

La aplicación de preparación de clases abarca los siguientes casos de uso: “Gestionar cursos”, “Gestionar sesiones”, “Gestionar tweets”, “Gestionar conceptos” y “Gestionar preguntas” así como el caso de uso “Consultar estadísticas”.

#### 4.2.1.1 Editores de contenidos

Los casos de uso de gestión están implementados dentro del componente de editores de contenidos, dicho componente engloba a cuatro componentes encargados de los procesos de alta, baja, modificación y consulta de los contenidos —cursos, sesiones, tweets, conceptos y preguntas— que forman parte de la aplicación.

#### 4.2.1.2 Gestor de conceptos

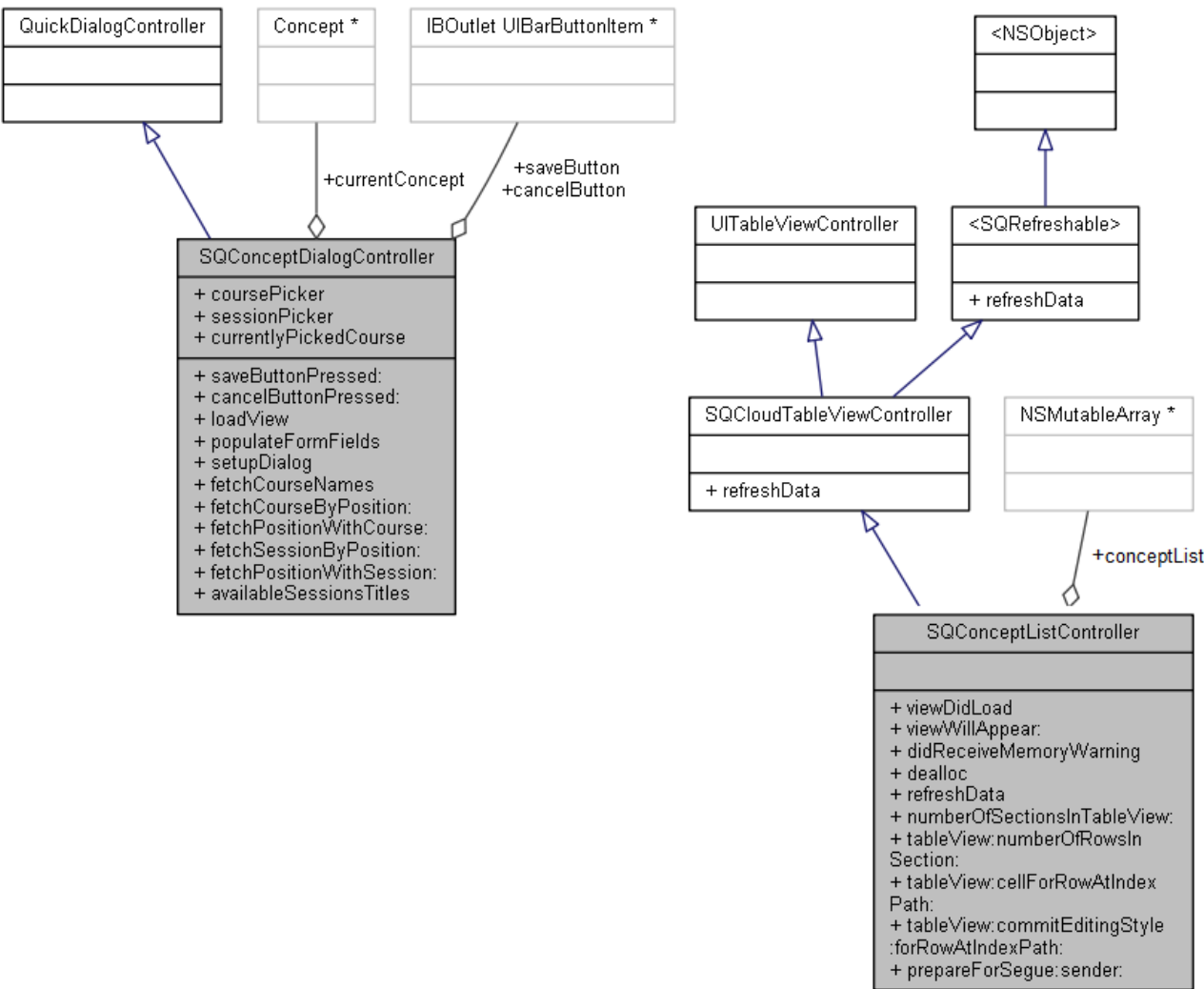


Ilustración 20: Detalle del componente 'Gestor de Conceptos'

La funcionalidad del componente ‘Gestor de conceptos’ está implementada en dos clases —señaladas en gris en el diagrama—, *SQConceptDialogController* que se encarga de proporcionar la funcionalidad de las operaciones de alta, baja y modificación de conceptos y la clase *SQConceptListController* que implementa la funcionalidad de listado (consulta) de conceptos almacenados en el sistema.

El resto de clases son clases de soporte que proporciona el sistema operativo (*UITableViewController*, *UIBarButtonItem*), clases de soporte del propio componente ‘Editor de contenidos’ (*SQRefreshable*, *SQCloudTableViewController*), librerías de terceros (*QuickDialog*) y clases que posibilitan la interacción con otros componentes, como la clase *Concept*, que pertenece al componente ‘Gestor de acceso a datos’ que alberga el modelo de la aplicación.

4.2.1.3 Gestor de sesiones

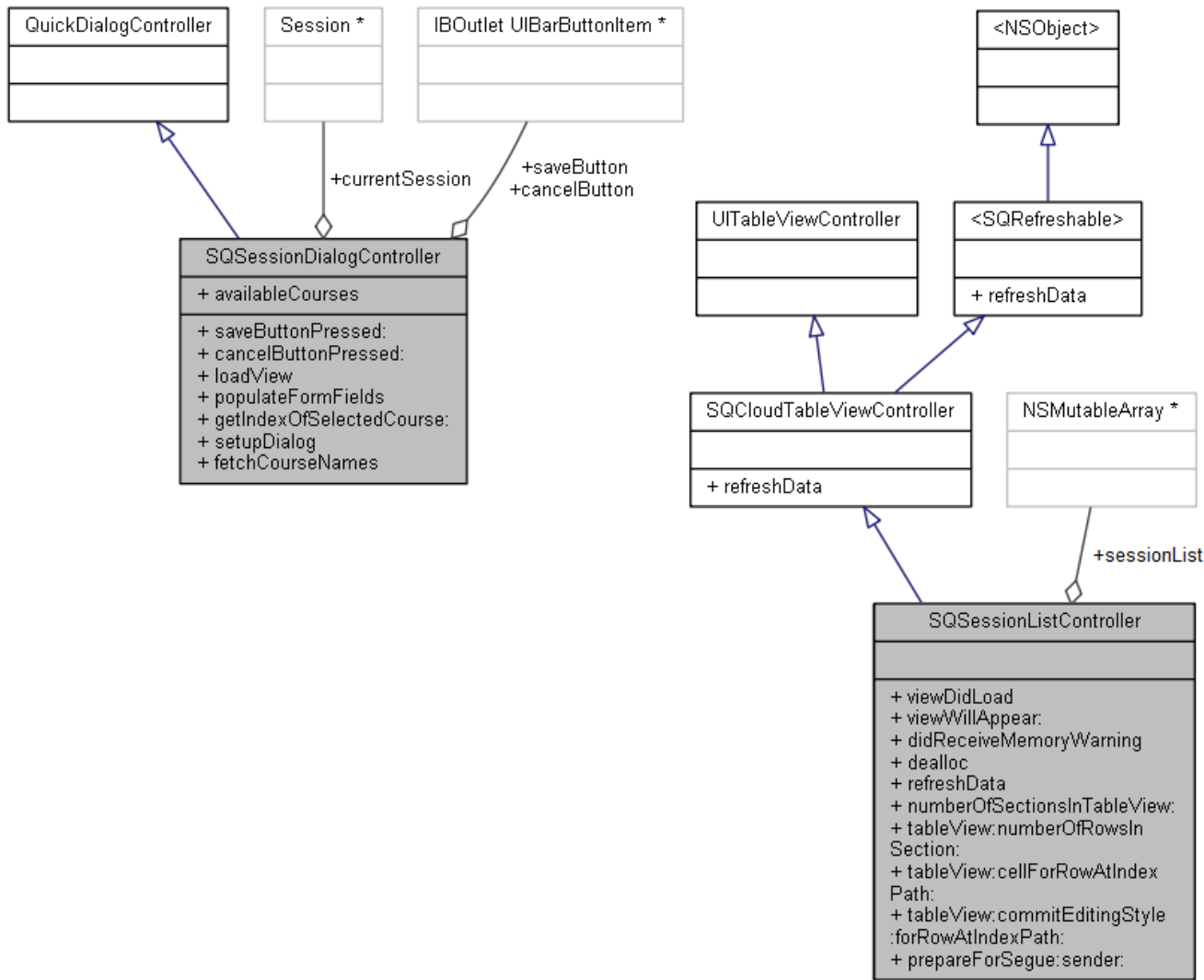


Ilustración 21: Detalle del componente ‘Gestor de sesiones’

La funcionalidad del componente ‘Gestor de sesiones’ está implementada en dos clases —señaladas en gris en el diagrama—, *SQSessionDialogController* que se encarga de proporcionar la funcionalidad de las operaciones de alta, baja y modificación de sesiones y la clase *SQSessionListController* que implementa la funcionalidad de listado (consulta) de sesiones almacenadas en el sistema.

El resto de clases son clases de soporte que proporciona el sistema operativo (*UITableViewController*, *UIBarButtonItem*), clases de soporte del propio componente ‘Editor de contenidos’ (*SQRefreshable*,

*SQCloudTableViewController*), librerías de terceros (*QuickDialog*) y clases que posibilitan la interacción con otros componentes, como la clase *Session*, que pertenece al componente ‘Gestor de acceso a datos’ que alberga el modelo de la aplicación.

4.2.1.4 Gestor de preguntas

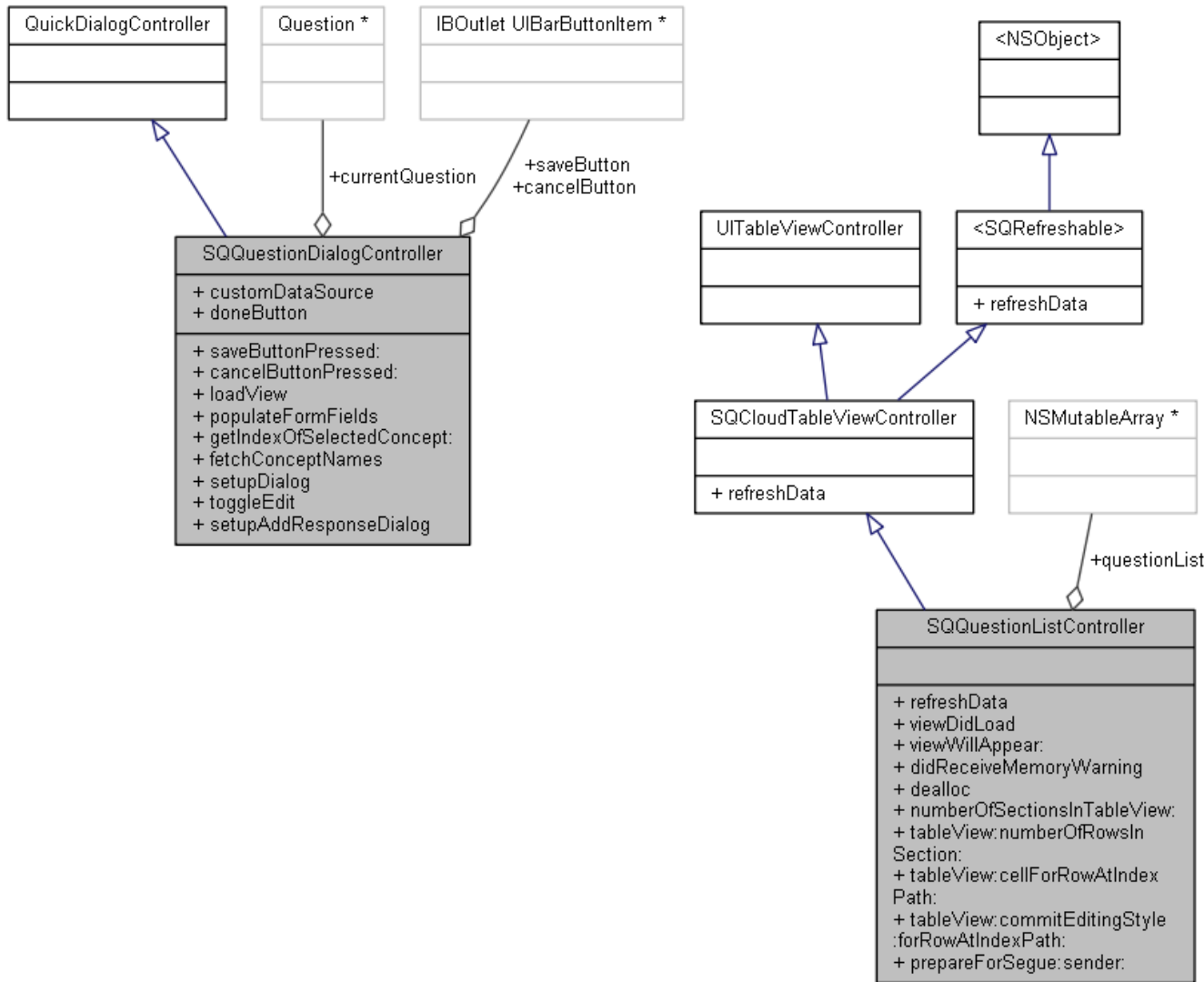


Ilustración 22: Detalle de componente ‘Gestor de preguntas’

La funcionalidad del componente ‘Gestor de preguntas’ está implementada en dos clases —señaladas en gris en el diagrama—, *SQQuestionDialogController* que se encarga de proporcionar la funcionalidad de las operaciones de alta, baja y modificación de preguntas y la clase *SQQuestionListController* que implementa la funcionalidad de listado (consulta) de preguntas almacenadas en el sistema.

El resto de clases son clases de soporte que proporciona el sistema operativo (*UITableViewController*, *UIBarButtonItem*), clases de soporte del propio componente ‘Editor de contenidos’ (*SQRefreshable*, *SQCloudTableViewController*), librerías de terceros (*QuickDialog*) y clases que posibilitan la interacción con otros componentes, como la clase *Session*, que pertenece al componente ‘Gestor de acceso a datos’ que alberga el modelo de la aplicación.

## 4.2.1.5 Gestor de cursos

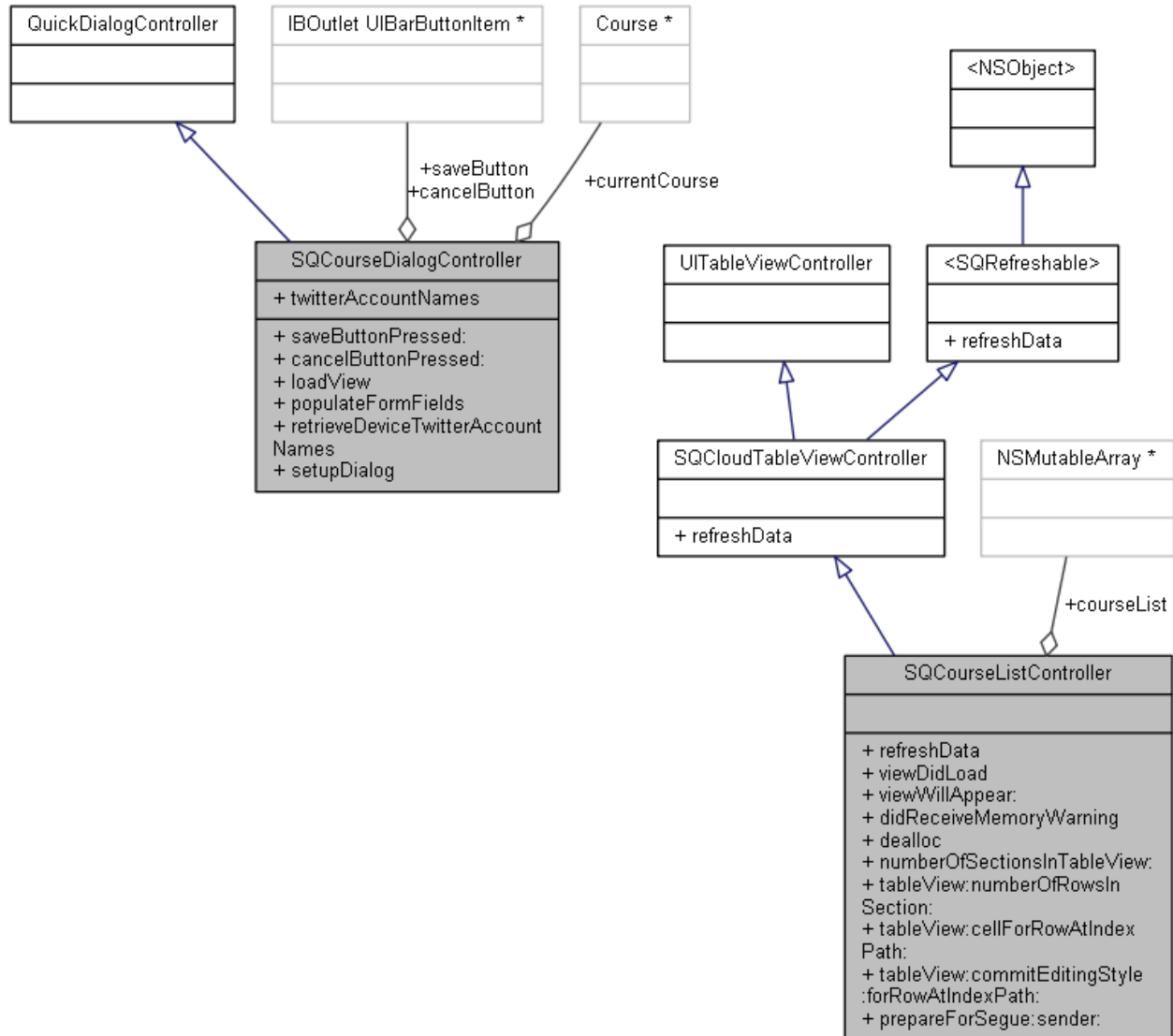


Ilustración 23: Detalle del componente 'Gestor de cursos'

La funcionalidad del componente "Gestor de cursos" está implementada en dos clases —señaladas en gris en el diagrama—, *SQCourseDialogController* que se encarga de proporcionar la funcionalidad de las operaciones de alta, baja y modificación de cursos y la clase *SQCourseListController* que implementa la funcionalidad de listado (consulta) de preguntas almacenadas en el sistema.

El resto de clases son clases de soporte que proporciona el sistema operativo (*UITableViewController*, *UIBarButtonItem*), clases de soporte del propio componente 'Editor de contenidos' (*SQRefreshable*, *SQCloudTableViewController*), librerías de terceros (*QuickDialog*) y clases que posibilitan la interacción con otros componentes, como la clase *Course*, que pertenece al componente 'Gestor de acceso a datos' que alberga el modelo de la aplicación.

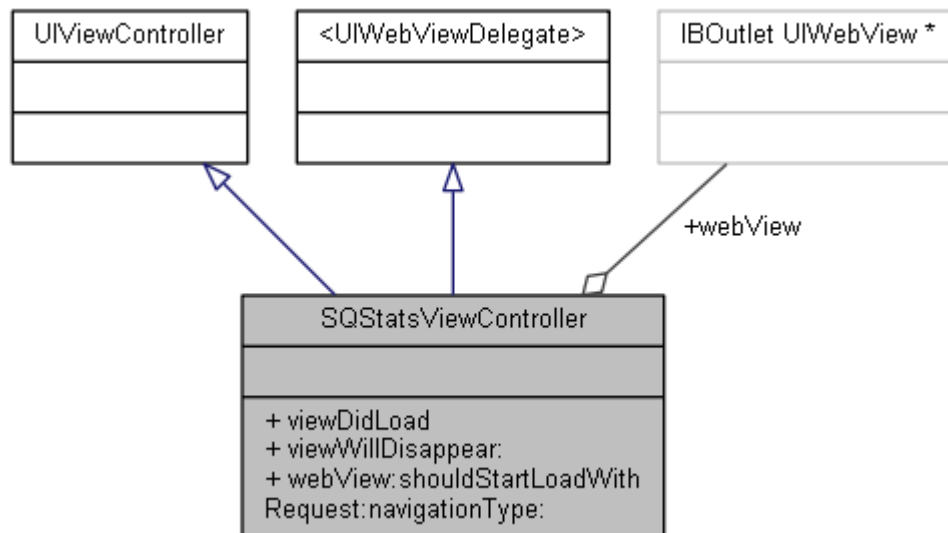
**4.2.1.6 Visor de estadísticas**

Ilustración 24: Detalle del componente 'Visor de estadísticas'

El componente visor de estadísticas está implementado como un visor web que accede a los servicios web de la aplicación y recupera una página de estadísticas generada al vuelo que está optimizada para dispositivos móviles para ser finalmente mostrada en el visor de la aplicación.



## 4.2.1.7 Gestor de acceso a datos

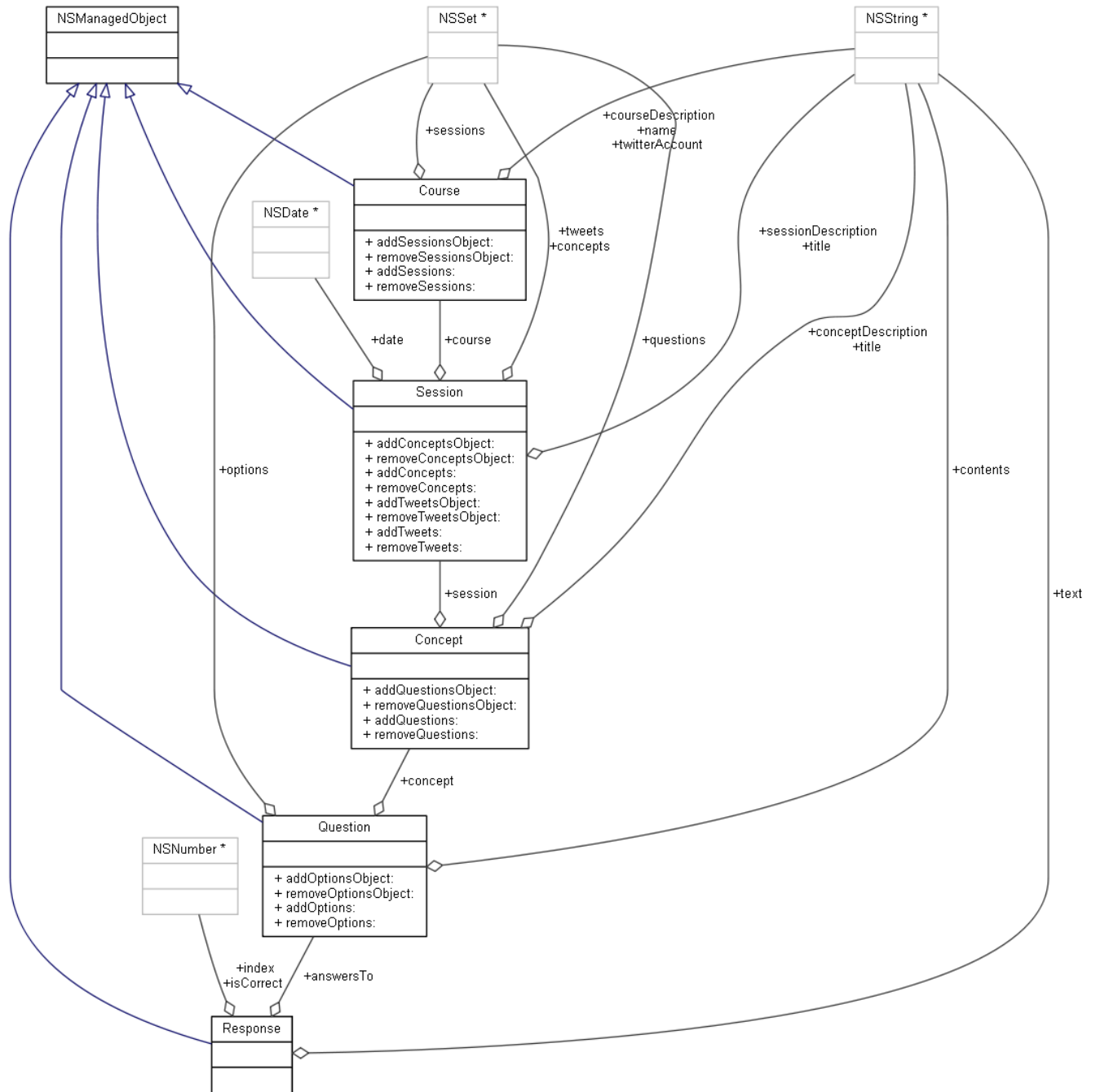


Ilustración 25: Detalle del componente 'Gestor de acceso a datos'

En el anterior diagrama se detalla la estructura del modelo de la aplicación y la interrelación entre cada uno de las clases que lo componen. La funcionalidad de persistencia de datos es llevada a cabo por el ancestro común de las clases del modelo, la clase *NSManagedObject*, proporcionada por el sistema operativo, mejorada y ampliada a través del sistema de categorías del lenguaje Objective-C por la librería externa MagicalRecord.

Las clases del modelo realizan operaciones de alta y baja que son delegadas a *NSManagedObject* para la persistencia de los datos.

### 4.2.2 Aplicación del alumno

La aplicación del alumno abarca los siguientes casos de uso: “Hacer checkin”, “Ver sesiones abiertas”, “Suscribirse a sesión” y “Responder preguntas”.

La aplicación del alumno antes de dar acceso al resto de componentes debe encargarse de registrar al usuario dentro de los servicios web del sistema, pues el resto de componentes de la aplicación dependen de dicho inicio de sesión para mostrar la información que es relevante a cada usuario.

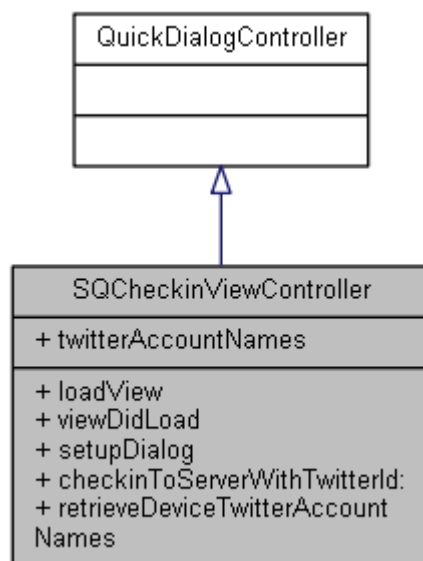


Ilustración 26: Controlador encargado del inicio de sesión de un cliente

El controlador encargado del inicio de sesión se encarga de recoger los nombres de cuentas twitter que hay en el dispositivo desde el que se inicia sesión para mostrárselos al usuario y que decida con qué usuario desea realizar el *check-in*.

Cuando el usuario en la vista da orden de realizar el *check-in* se llama al método *checkinToServerWithTwitterId: (NSString\*) twitterId*, se realiza la llamada pertinente al servicio web y el usuario queda registrado en el servidor.

#### 4.2.2.1 Navegador de sesiones

Tras realizar el inicio de sesión el usuario se encuentra con el navegador de sesiones. El navegador de sesiones lista las sesiones que hay abiertas en un determinado instante en los servicios web de la aplicación, el usuario puede elegir a través de la interacción con la vista de este componente a qué sesiones desea suscribirse para recibir las preguntas que se planteen dentro de dicha sesión.

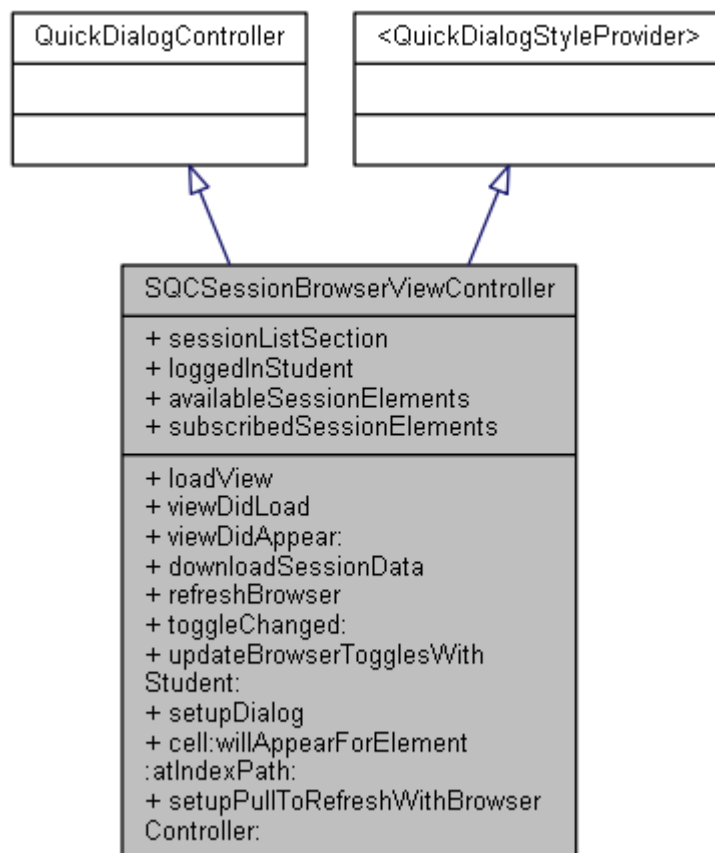


Ilustración 27: Detalle del componente 'Navegador de sesiones'

El controlador del navegador de sesiones es el encargado de actualizar la correspondiente vista de sesiones disponibles, mostrando las sesiones que a las que un usuario puede suscribirse y a las que ya está suscrito. El controlador también ha de tener la capacidad de mantener consistente el estado de las suscripciones con el servidor, es decir, si un usuario desea suscribirse a una sesión y sucede un error durante el proceso la sesión deberá volver a su estado original: no suscrita. En caso contrario, para sesiones ya suscritas y que de las que se desee cancelar la suscripción, el proceso explicado anteriormente también es aplicable pero en sentido inverso.

#### 4.2.2.2 Inbox de preguntas

Este componente realiza la función de bandeja de llegada de preguntas para un alumno, listándolas y mostrando cuáles han sido ya vistas y cuáles no. Pulsando en cualquiera de las preguntas de la lista se invoca al visor de preguntas donde el alumno puede leer con detalle la pregunta y responderla.

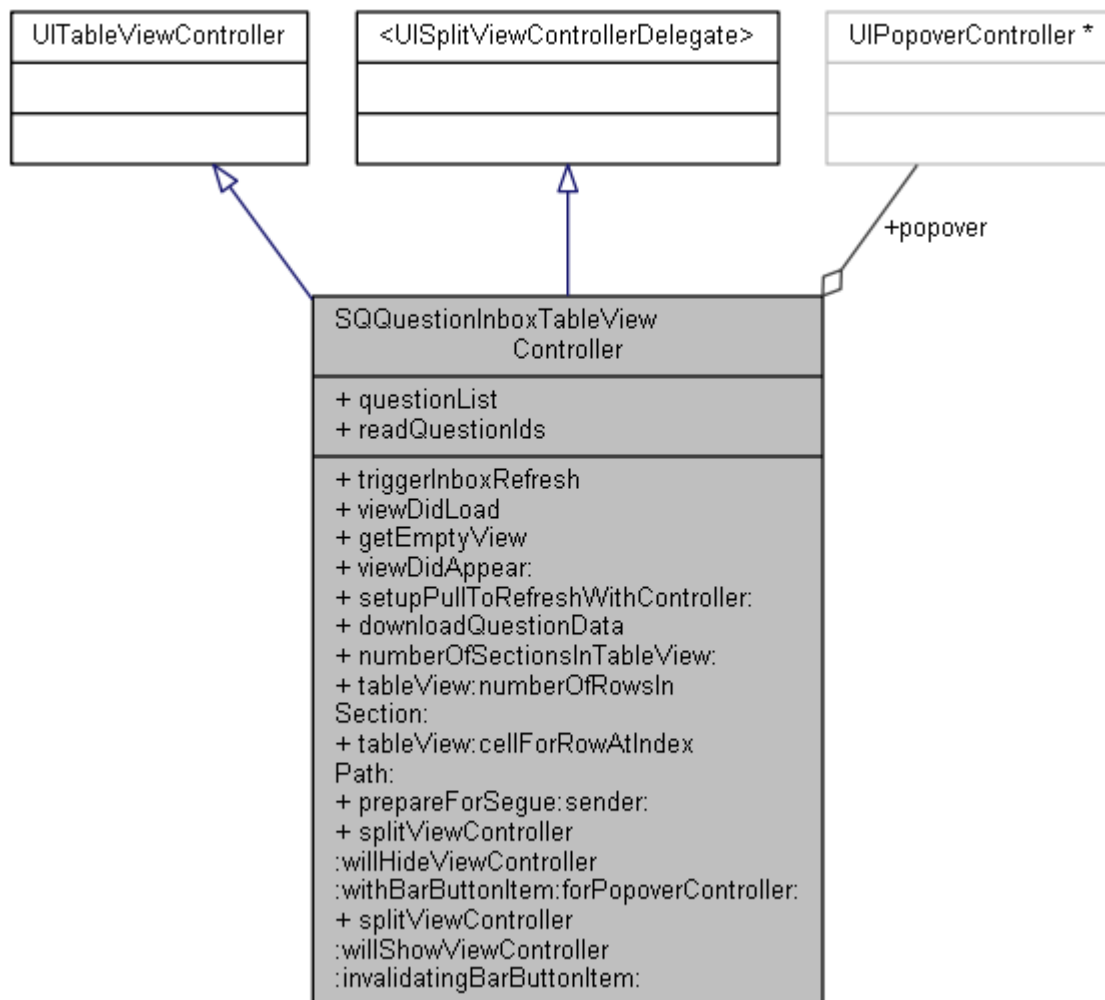


Ilustración 28: Detalle del componente 'Inbox de preguntas'

### 4.2.3 Servicios Web

El componente ‘Servicios Web’ contiene los servicios sobre los que se asientan el resto de aplicaciones. En este apartado se detallará únicamente el componente ‘Cuestionarios’ pues los componentes ‘Cloud Data’ y ‘Apple Push Notification Service’ son externos, proporcionados por terceros y ajenos por tanto a este proyecto.

#### 4.2.3.1 Cuestionarios

El componente ‘Cuestionarios’ está formado por un servicio web desarrollado usando *Play! Framework* siguiendo el patrón MVC. El servicio web expone una serie de métodos a los consumidores con los que interactuar a través de una arquitectura REST. Dichos métodos serán detallados en la sección *Servicios Web: Especificación de la API REST del servicio de cuestionarios*.

Para comunicar las aplicaciones de usuarios y profesores de la plataforma iOS se ha desarrollado una librería que implementa los métodos del servicio web para ser empleada por las aplicaciones iOS que necesiten acceder a los servicios del componente.

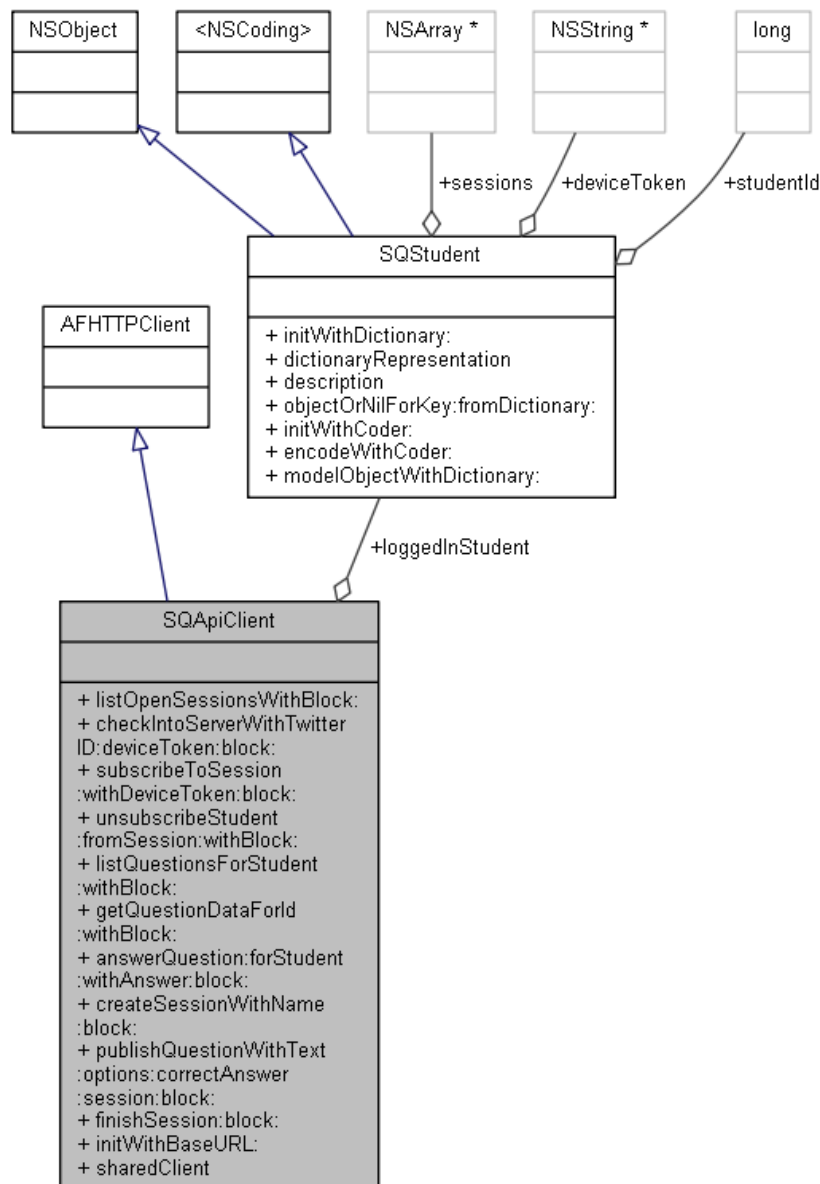


Ilustración 29: Detalle de la librería de abstracción del API REST para iOS

## 4.3 Servicios Web: Especificación de la API REST del servicio de cuestionarios

En este apartado se procederá a documentar los métodos de los que consta el servicio web de cuestionarios y que deben emplear las aplicaciones para el intercambio de preguntas y respuestas entre profesores y alumnos.

### 4.3.1 Sesión

Operaciones sobre una sesión de clase

#### 4.3.1.1 Crear sesión

Crea una nueva sesión a la que los alumnos pueden suscribirse.

##### 4.3.1.1.1 Solicitud

Método	URL	
POST	/session/create	
Método	Parámetros	Tipo de datos
POST	title	string

Tabla 43: Datos de llamada del método 'Crear sesión'

##### 4.3.1.1.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 1,   "title": "Programación móvil en iOS",   "sessionDate": 1380831317819,   "sessionClosed": false,   "questions": [] }</pre>

Tabla 44: Códigos de respuesta del método 'Crear sesión'

#### 4.3.1.2 Listar sesiones abiertas

Lista las sesiones que hay en curso en el instante de la solicitud.

##### 4.3.1.2.1 Solicitud

Método	URL
GET	/session/listopen

Tabla 45: Datos de llamada del método 'Listar sesiones abiertas'

##### 4.3.1.2.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>[   {     "id": 1,     "title": "Programación móvil en iOS",     "sessionDate": 1380831317819,     "sessionClosed": false,     "questions": []   },   {     "id": 2,     "title": "Introducción a Android",     "sessionDate": 1380831858466,     "sessionClosed": false,     "questions": []   } ]</pre>

Tabla 46: Códigos de respuesta del método 'Listar sesiones abiertas'

#### 4.3.1.3 Consultar sesión

Crea una nueva sesión a la que los alumnos pueden suscribirse.

##### 4.3.1.3.1 Solicitud

Método	URL	
GET	/session/{sessionId}	
Método	Parámetros	Tipo de datos
GET	sessionId	long

Tabla 47: Datos de llamada del método 'Consultar sesión'

##### 4.3.1.3.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 1,   "title": "Programación móvil en iOS",   "sessionDate": 1380831317819,   "sessionClosed": false,   "questions": [     {       "id": 1,       "questionText": "El lenguaje en el que se programa iOS es... ",       "answerOptions": [         {           "id": 1,           "answerText": "Java"         },         {           "id": 2,           "answerText": "Objective-C"         },         {           "id": 3,           "answerText": "Klingon"         }       ]     }   ] }</pre>
404	<pre>{"error": "No session found with id: &lt;sessionId&gt;"}</pre>

Tabla 48: Códigos de respuesta del método 'Consultar sesión'

#### 4.3.1.4 Finalizar sesión

Finaliza una sesión en curso quedando inactiva para los alumnos, evitando que puedan seguir contestando cuestiones una vez el profesor decida que la sesión ha expirado. Una vez finaliza una sesión no se permite la suscripción de nuevos alumnos a la misma.

Las sesiones finalizadas no se eliminan, sólo se archivan.

##### 4.3.1.4.1 Solicitud

Método	URL	
POST	/session/finish/{sessionId}	
Método	Parámetros	Tipo de datos
POST	sessionId	long

Tabla 49: Datos de llamada del método 'Finalizar sesión'

##### 4.3.1.4.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 1,   "title": "Programación móvil en iOS",   "sessionDate": 1380831317819,   "sessionClosed": true,   "questions": [] }</pre>

<b>404</b>	<code>{"error": "No session found with id: &lt;sessionId&gt;"}</code>
------------	---

Tabla 50: Códigos de respuesta del método 'Finalizar sesión'

#### 4.3.1.5 Eliminar sesión

Finaliza una sesión en curso quedando inactiva para los alumnos.

##### 4.3.1.5.1 Solicitud

Método	URL	
<b>DELETE</b>	/session/delete/{sessionId}	
Método	Parámetros	Tipo de datos
<b>DELETE</b>	sessionId	long

Tabla 51: Datos de llamada del método 'Eliminar sesión'

##### 4.3.1.5.2 Respuesta

Código	Ejemplo de respuesta
<b>200</b>	<pre>{   "id": 1,   "title": "Programación móvil en iOS",   "sessionDate": 1380831317819,   "sessionClosed": true,   "questions": [] }</pre>
<b>404</b>	<code>{"error": "No session found with id: &lt;sessionId&gt;"}</code>

Tabla 52: Códigos de respuesta del método 'Eliminar sesión'

#### 4.3.1.6 Publicar pregunta

Lanza una pregunta dentro de una sesión para que sea recibida por los alumnos suscritos a la misma.

##### 4.3.1.6.1 Solicitud

Método	URL	
<b>POST</b>	/session/{sessionId}/publish/{questionId}	
Método	Parámetros	Tipo de datos
<b>POST</b>	sessionId	long
<b>POST</b>	questionId	long

Tabla 53: Datos de llamada del método 'Eliminar sesión'

##### 4.3.1.6.2 Respuesta

Código	Ejemplo de respuesta
<b>200</b>	<pre>{   "id": 1,   "questionText": "El lenguaje en el que se programa iOS es... ",   "answerOptions": [     {       "id": 1,       "answerText": "Java"     },     {       "id": 2,       "answerText": "Objective-C"     },     {       "id": 3,       "answerText": "Klingon"     }   ] }</pre>
<b>404</b>	<code>{"error": "No session found with id: &lt;sessionId&gt;"}</code>

Tabla 54: Códigos de respuesta del método 'Eliminar sesión'



### 4.3.2 Alumnos

En esta sección se especifican las operaciones que pueden realizar los alumnos y que pueden realizarse sobre ellos.

#### 4.3.2.1 Check-in

Este método permite que un alumno inicie sesión en el sistema, permitiéndole suscribirse a asignaturas, recibir preguntas y enviar respuestas identificables por los profesores.

##### 4.3.2.1.1 Solicitud

Método	URL	
POST	/student/checkin	
Método	Parámetros	Tipo de datos
POST	deviceToken	string
POST	twitterId	string

Tabla 55: Datos de llamada del método 'Check-in'

##### 4.3.2.1.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 1,   "twitterId": "@CUADERNOUC3M",   "deviceToken": "7ikhgfds89hfsdfgydf84wq3fldhs02w47",   "sessions": [] }</pre>

Tabla 56: Códigos de respuesta del método 'Check-in'

#### 4.3.2.2 Suscribirse a una sesión

Este método permite que un alumno se suscriba a una sesión y pueda recibir en su dispositivo las preguntas que un profesor lance dentro de una sesión de clase.

##### 4.3.2.2.1 Solicitud

Método	URL	
POST	/student/{sessionId}/subscribe	
Método	Parámetros	Tipo de datos
POST	sessionId	long
POST	deviceToken	string

Tabla 57: Datos de llamada del método 'Suscribirse a una sesión'

##### 4.3.2.2.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 1,   "twitterId": "@CUADERNOUC3M",   "deviceToken": "7ikhgfds89hfsdfgydf84wq3fldhs02w47",   "sessions": [     {       "id": 1,       "title": "Programación móvil en iOS",       "sessionDate": 1380878947821,       "sessionClosed": false,       "questions": []     }   ] }</pre>
404	<pre>{"error": "Impossible to check into the session with id {sessionId} because it does not exist"}</pre>

Tabla 58: Códigos de respuesta del método 'Suscribirse a una sesión'

#### 4.3.2.3 Darse de baja de una sesión

Este método permite que un alumno se dé de baja de una sesión y pueda dejar de recibir en su dispositivo las preguntas que un profesor lance dentro de una sesión de clase.

##### 4.3.2.3.1 Solicitud

Método	URL	
DELETE	/student/{studentId}/unsubscribe/{sessionId}	
Método	Parámetros	Tipo de datos
DELETE	sessionId	long
DELETE	studentId	long

Tabla 59: Datos de llamada del método 'Darse de baja de una sesión'

##### 4.3.2.3.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 1,   "twitterId": "@CUADERNOUC3M",   "deviceToken": "7ikhgfs89hfsdfgydf84wq3fldhs02w47",   "sessions": [] }</pre>
404	<pre>{"error": "Impossible to unsubscribe from the session with id {sessionId} because it does not exist"}</pre>
404	<pre>{"error": "The student with id: {studentId} does not exist and can't be unsubscribed"}</pre>

Tabla 60: Códigos de respuesta del método 'Darse de baja de una sesión'

#### 4.3.2.4 Listar preguntas de un alumno

Este método permite listar un histórico con todas las preguntas que un alumno ha respondido.

##### 4.3.2.4.1 Solicitud

Método	URL	
GET	/student/{studentId}/questions	
Método	Parámetros	Tipo de datos
GET	studentId	long

Tabla 61: Datos de llamada del método 'Listar preguntas de un alumno'

##### 4.3.2.4.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>[   {     "id": 1,     "questionText": "El lenguaje en el que se programa iOS es... ",     "answerOptions": [       {         "id": 1,         "answerText": "Java"       },       {         "id": 2,         "answerText": "Objective-C"       }     ]   } ]</pre>
404	<pre>{"error": "No student found with id {studentId}"}</pre>

Tabla 62: Códigos de respuesta del método 'Listar preguntas de un alumno'

#### 4.3.2.5 Obtener un alumno

Este método permite obtener los datos de un alumno ya existente en el sistema

##### 4.3.2.5.1 Solicitud

Método	URL	
GET	/student/{studentId}	
Método	Parámetros	Tipo de datos
GET	studentId	long

Tabla 63: Datos de llamada del método 'Obtener un alumno'

##### 4.3.2.5.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 1,   "twitterId": "@CUADERNOUC3M",   "deviceToken": "7ikhgfs89hfsdfgydf84wq3fldhs02w47",   "sessions": [     {       "id": 1,       "title": "Programación móvil en iOS",       "sessionDate": 1380878947821,       "sessionClosed": false,       "questions": [         {           "id": 1,           "questionText": "El lenguaje de iOS es... ",           "answerOptions": [             {               "id": 1,               "answerText": "Java"             },             {               "id": 2,               "answerText": "Objective-C"             }           ]         }       ]     }   ] }</pre>
404	<pre>{"error": "No student found with id {studentId}"}</pre>

Tabla 64: Códigos de respuesta del método 'Obtener un alumno'

#### 4.3.2.6 Borrar un alumno

Este método permite eliminar los datos de un alumno ya existente en el sistema

##### 4.3.2.6.1 Solicitud

Método	URL	
DELETE	/student/{studentId}	
Método	Parámetros	Tipo de datos
DELETE	studentId	long

Tabla 65: Datos de llamada del método 'Borrar un alumno'

##### 4.3.2.6.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 1,   "twitterId": "@CUADERNOUC3M",</pre>

	<pre>"deviceToken": "71khgfds89hfsdfgydf84wq3fldhs02w47", "sessions": [   {     "id": 1,     "title": "Programación móvil en iOS",     "sessionDate": 1380878947821,     "sessionClosed": false,     "questions": [       {         "id": 1,         "questionText": "El lenguaje de iOS es...",         "answerOptions": [           {             "id": 1,             "answerText": "Java"           },           {             "id": 2,             "answerText": "Objective-C"           }         ]       }     ]   } ]</pre>
404	<pre>{"error": "No student found with id {studentId}"}</pre>

Tabla 66: Códigos de respuesta del método 'Borrar un alumno'

4.3.3 Preguntas

En esta sección se especifican las operaciones que pueden realizarse con las preguntas

4.3.3.1 Crear pregunta

Este método permite crear una nueva pregunta para ser publicada.

4.3.3.1.1 Solicitud

Método	URL	
POST	/question/create	
Método	Parámetros	Tipo de datos
POST	text	string
POST	correctOption	integer
POST	options	string[]

Tabla 67: Datos de llamada del método 'Crear pregunta'

4.3.3.1.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 2,   "questionText": "La captura de requisitos debe realizarla...",   "answerOptions": [     {       "id": 4,       "answerText": "El analista"     },     {       "id": 5,       "answerText": "El diseñador"     }   ] }</pre>

Tabla 68: Códigos de respuesta del método 'Crear pregunta'

4.3.3.2 Obtener pregunta

Este método permite recuperar la información sobre una pregunta existente

4.3.3.2.1 Solicitud

Método	URL	
GET	/question/{questionId}	
Método	Parámetros	Tipo de datos
GET	questionId	long

Tabla 69: Datos de llamada del método 'Obtener pregunta'

4.3.3.2.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 2,   "questionText": "La captura de requisitos debe realizarla...",   "answerOptions": [     {       "id": 4,       "answerText": "El analista"     },     {       "id": 5,       "answerText": "El diseñador"     }   ] }</pre>
404	<pre>{"error": "No question found with id {questionId}"}</pre>

Tabla 70: Códigos de respuesta del método 'Obtener pregunta'

4.3.3.3 Borrar pregunta

Este método permite eliminar la información sobre una pregunta existente

4.3.3.3.1 Solicitud

Método	URL	
DELETE	/question/{questionId}	
Método	Parámetros	Tipo de datos
DELETE	questionId	long

Tabla 71: Datos de llamada del método 'Borrar pregunta'

4.3.3.3.2 Respuesta

Código	Ejemplo de respuesta
200	<pre>{   "id": 2,   "questionText": "La captura de requisitos debe realizarla...",   "answerOptions": [     {       "id": 4,       "answerText": "El analista"     },     {       "id": 5,       "answerText": "El diseñador"     }   ] }</pre>
404	<pre>{"error": "Impossible to delete, no question found with id {questionId}"}</pre>

Tabla 72: Códigos de respuesta del método 'Borrar pregunta'

#### 4.3.3.4 Responder pregunta

Este método permite enviar una respuesta a una pregunta existente

##### 4.3.3.4.1 Solicitud

Método	URL	
POST	/question/{questionId}/answer/{answerId}/student/{studentId}	
Método	Parámetros	Tipo de datos
POST	questionId	long
POST	answerId	long
POST	studentId	long

Tabla 73: Datos de llamada del método 'Responder pregunta'

##### 4.3.3.4.2 Respuesta

Código	Ejemplo de respuesta
200	{ "id":2, "answerText":"Objective-C" }
404	{"error": "The question with id: {questionId} does not exist"}
404	{"error": "The answer with id: {answerId} does not exist"}
404	{"error": "The student with id: {studentId} does not exist"}

Tabla 74: Códigos de respuesta del método 'Responder pregunta'

## 4.4 Diagramas de secuencia

En este apartado se describirá el funcionamiento de las aplicaciones a través de diagramas de secuencia en lenguaje UML, detallando así las interacciones entre los distintos componentes de las mismas.

Los diagramas se han estructurado siguiendo la especificación de casos de uso, es decir, cada diagrama de secuencia engloba la funcionalidad contenida en uno o varios casos de uso.

### 4.4.1 Gestión de cursos

La secuencia gestión de cursos se divide en tres acciones elementales: listar cursos, crear un curso, edición de datos y borrado de un curso.

1. *UIApplicationMain()*: El usuario abre la aplicación y este mensaje desencadena la activación de la aplicación por parte del sistema operativo.
  - 1.1. *applicationDidBecomeActive*: La aplicación está activa y se procede a la carga de la pantalla principal.
    - 1.1.1. *viewDidLoad*: La pantalla principal ya ha sido cargada y está lista para ser mostrada al usuario. Se muestra la pantalla al usuario y este ya puede interactuar con la aplicación.
2. *tableView:didSelectRowAtIndexPath*: El usuario ha seleccionado la opción de gestión de cursos en la pantalla principal y ha desencadenado el evento de 'opción seleccionada'.
  - 2.1. La pantalla principal de la aplicación notifica a *UIKit* (el gestor de interfaz de usuario del SO) a través del método *performSegueWithIdentifier*: que ha sido seleccionada una opción de navegación hacia el gestor de cursos y que debe instanciarlo y mostrarlo.
    - 2.1.1. *UIKit* ha instanciado el controlador de destino, este se ha cargado y justo en ese instante se ejecuta el método *viewWillAppear*: el cual indica que la vista asociada a este controlador está a punto de mostrarse. Este momento es la última oportunidad para recuperar los datos existentes para mostrarlos al usuario en cuanto aparezca la vista en pantalla.
      - 2.1.1.1. *SQLCourseListController* invoca a su propio método *refreshData*: para cargar los datos que se van a listar en pantalla. El controlador se muestra en pantalla listo para que el usuario interactúe con él.
3. Para introducir nuevos datos el usuario debe acceder hasta la pantalla de listado de los mismos, una allí el usuario invoca la pantalla de adición/edición de cursos a través de *UIKit*, pulsando en el botón correspondiente.
  - 3.1. *UIKit* invoca al método *presentViewController:animated*: que presenta al usuario el diálogo de adición/edición de cursos. El usuario introduce los datos.
4. El usuario invoca desencadena el evento *saveButtonPressed*: pulsando en el botón de guardado de datos cuando ha finalizado
  - 4.1. El diálogo de adición/edición *SQLCourseDialogController* el método *postNotificationName*: de *UIKit* para enviar la notificación de actualización a los consumidores de la información suscritos.
    - 4.1.1. *UIKit* invoca el método *refreshData*: de la clase *SQLCourseListController*, suscrita a recibir notificaciones por parte del SO sobre cambios en la información que le interesa.
5. Para eliminar un curso, el usuario pone en modo de edición la lista de cursos a través de *setEditing:animated*:
6. El usuario le dice a *UIKit* que quiere borrar un elemento de la lista
  - 6.1. *UIKit* invoca a *commitEditingStyle:forRowAtIndexPath*: para que *SQLCourseListController* efectúe el borrado de los datos.

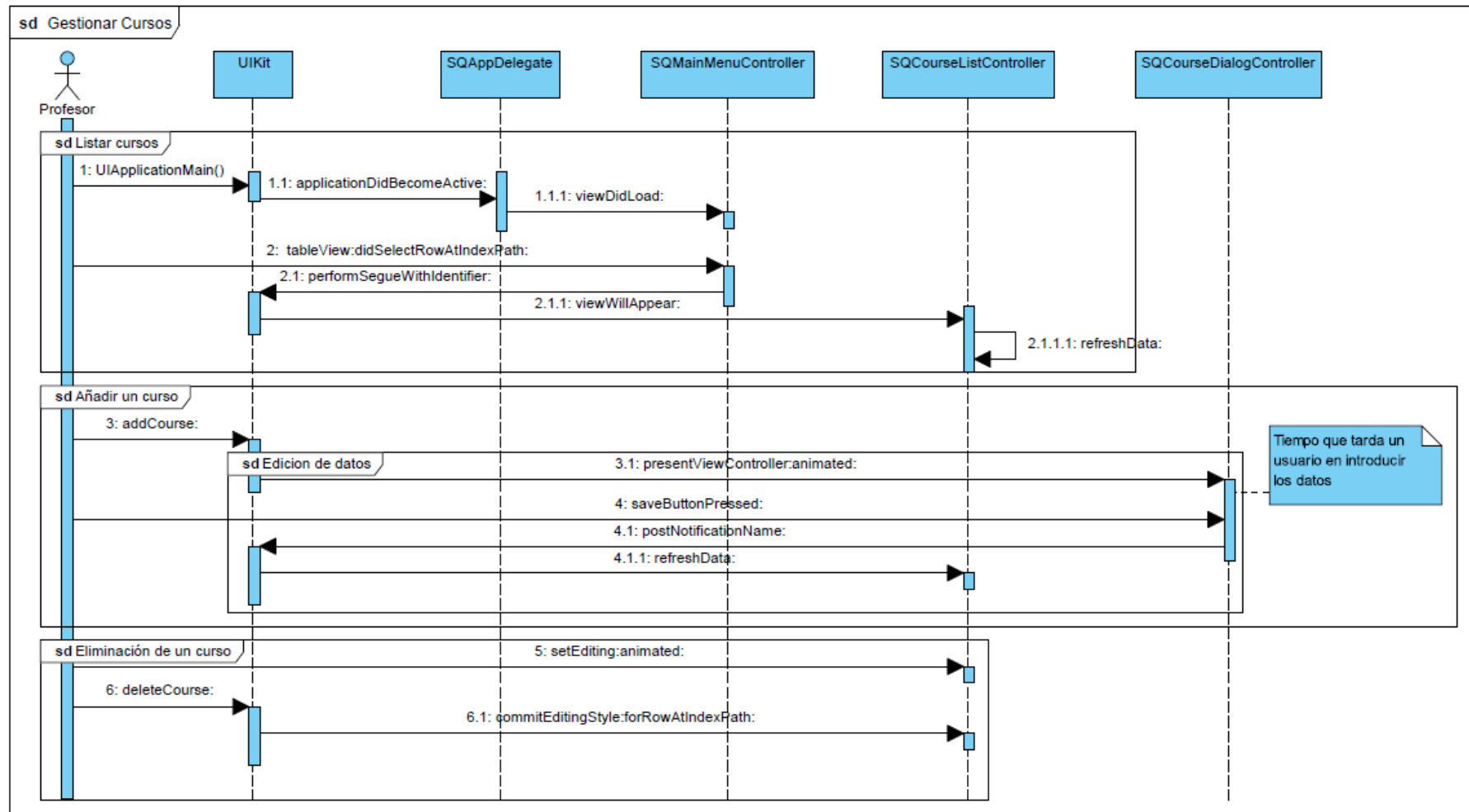


Ilustración 30: Diagrama de secuencia 'Gestionar Cursos'



#### 4.4.2 Gestión de sesiones

La secuencia gestión de sesiones se divide en tres acciones elementales: listar sesiones, crear una sesión, edición de datos y borrado de una sesión.

1. *UIApplicationMain()*: El usuario abre la aplicación y este mensaje desencadena la activación de la aplicación por parte del sistema operativo.
  - 1.1. *applicationDidBecomeActive*: La aplicación está activa y se procede a la carga de la pantalla principal.
    - 1.1.1. *viewDidLoad*: La pantalla principal ya ha sido cargada y está lista para ser mostrada al usuario. Se muestra la pantalla al usuario y este ya puede interactuar con la aplicación.
2. *tableView:didSelectRowAtIndexPath*: El usuario ha seleccionado la opción de gestión de sesiones en la pantalla principal y ha desencadenado el evento de 'opción seleccionada'.
  - 2.1. La pantalla principal de la aplicación notifica a *UIKit* (el gestor de interfaz de usuario del SO) a través del método *performSegueWithIdentifier*: que ha sido seleccionada una opción de navegación hacia el gestor de sesiones y que debe instanciarlo y mostrarlo.
    - 2.1.1. *UIKit* ha instanciado el controlador de destino, este se ha cargado y justo en ese instante se ejecuta el método *viewWillAppear*: el cual indica que la vista asociada a este controlador está a punto de mostrarse. Este momento es la última oportunidad para recuperar los datos existentes para mostrarlos al usuario en cuanto aparezca la vista en pantalla.
      - 2.1.1.1. *SQLSessionListController* invoca a su propio método *refreshData*: para cargar los datos que se van a listar en pantalla. El controlador se muestra en pantalla listo para que el usuario interactúe con él.
3. Para introducir nuevos datos el usuario debe acceder hasta la pantalla de listado de los mismos, una allí el usuario invoca la pantalla de adición/edición de sesiones a través de *UIKit*, pulsando en el botón correspondiente.
  - 3.1. *UIKit* invoca al método *presentViewController:animated*: que presenta al usuario el diálogo de adición/edición de sesiones. El usuario introduce los datos.
4. El usuario invoca desencadena el evento *saveButtonPressed*: pulsando en el botón de guardado de datos cuando ha finalizado
  - 4.1. El diálogo de adición/edición *SQLSessionDialogController* el método *postNotificationName*: de *UIKit* para enviar la notificación de actualización a los consumidores de la información suscritos.
    - 4.1.1. *UIKit* invoca el método *refreshData*: de la clase *SQLSessionListController*, suscrita a recibir notificaciones por parte del SO sobre cambios en la información que le interesa.
5. Para eliminar una sesión, el usuario pone en modo de edición la lista de sesiones a través de *setEditing:animated*:
6. El usuario le dice a *UIKit* que quiere borrar un elemento de la lista
  - 6.1. *UIKit* invoca a *commitEditingStyle:forRowAtIndexPath*: para que *SQLSessionListController* efectúe el borrado de los datos.

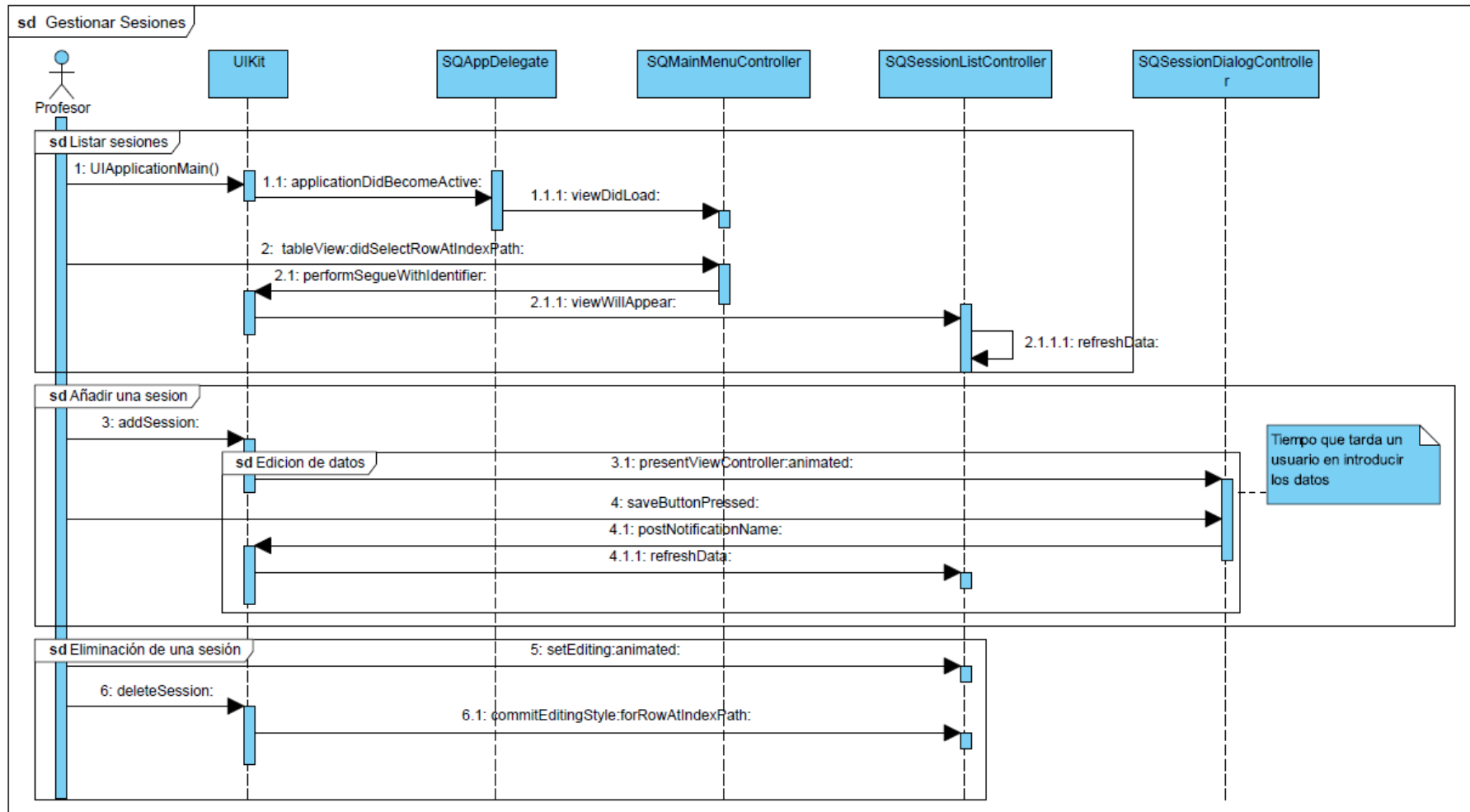


Ilustración 31: Diagrama de secuencia 'Gestionar Sesiones'

### 4.4.3 Gestión de preguntas

La secuencia gestión de preguntas se divide en tres acciones elementales: listar preguntas, crear una pregunta, edición de datos y borrado de preguntas.

1. *UIApplicationMain()*: El usuario abre la aplicación y este mensaje desencadena la activación de la aplicación por parte del sistema operativo.
  - 1.1. *applicationDidBecomeActive*: La aplicación está activa y se procede a la carga de la pantalla principal.
    - 1.1.1. *viewDidLoad*: La pantalla principal ya ha sido cargada y está lista para ser mostrada al usuario. Se muestra la pantalla al usuario y este ya puede interactuar con la aplicación.
2. *tableView:didSelectRowAtIndexPath*: El usuario ha seleccionado la opción de gestión de preguntas en la pantalla principal y ha desencadenado el evento de 'opción seleccionada'.
  - 2.1. La pantalla principal de la aplicación notifica a *UIKit* (el gestor de interfaz de usuario del SO) a través del método *performSegueWithIdentifier*: que ha sido seleccionada una opción de navegación hacia el gestor de preguntas y que debe instanciarlo y mostrarlo.
    - 2.1.1. *UIKit* ha instanciado el controlador de destino, este se ha cargado y justo en ese instante se ejecuta el método *viewWillAppear*: el cual indica que la vista asociada a este controlador está a punto de mostrarse. Este momento es la última oportunidad para recuperar los datos existentes para mostrarlos al usuario en cuanto aparezca la vista en pantalla.
      - 2.1.1.1. *SQQuestionListController* invoca a su propio método *refreshData*: para cargar los datos que se van a listar en pantalla. El controlador se muestra en pantalla listo para que el usuario interactúe con él.
3. Para introducir nuevos datos el usuario debe acceder hasta la pantalla de listado de los mismos, una allí el usuario invoca la pantalla de adición/edición de sesiones a través de *UIKit*, pulsando en el botón correspondiente.
  - 3.1. *UIKit* invoca al método *presentViewController:animated*: que presenta al usuario el diálogo de adición/edición de preguntas. El usuario introduce los datos.
4. El usuario invoca desencadena el evento *saveButtonPressed*: pulsando en el botón de guardado de datos cuando ha finalizado
  - 4.1. El diálogo de adición/edición *SQQuestionDialogController* el método *postNotificationName*: de *UIKit* para enviar la notificación de actualización a los consumidores de la información suscritos.
    - 4.1.1. *UIKit* invoca el método *refreshData*: de la clase *SQQuestionListController*, suscrita a recibir notificaciones por parte del SO sobre cambios en la información que le interesa.
5. Para eliminar una sesión, el usuario pone en modo de edición la lista de sesiones a través de *setEditing:animated*:
6. El usuario le dice a *UIKit* que quiere borrar un elemento de la lista
  - 6.1. *UIKit* invoca a *commitEditingStyle:forRowAtIndexPath*: para que *SQQuestionListController* efectúe el borrado de los datos.

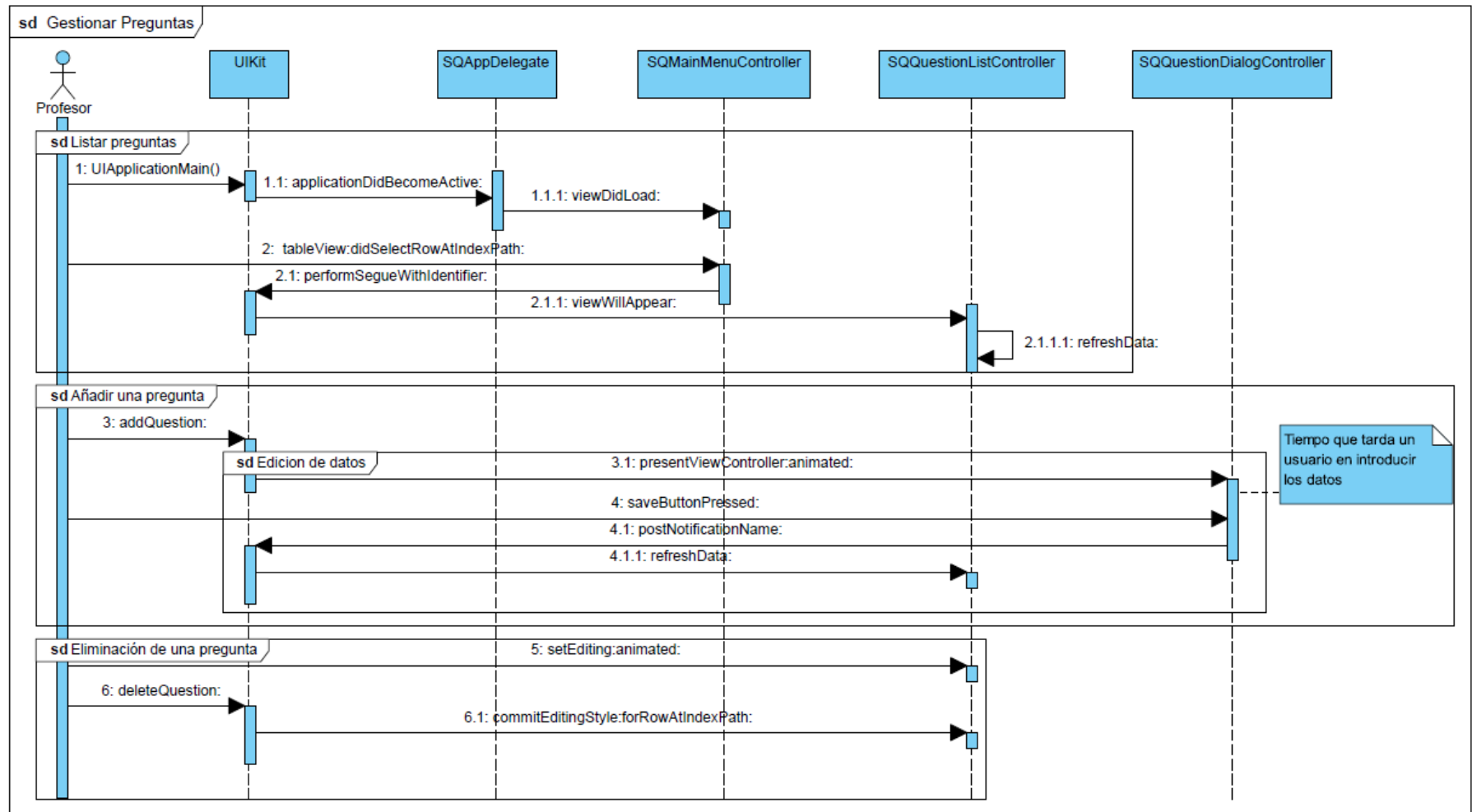


Ilustración 32: Diagrama de secuencia 'Gestionar Preguntas'

#### 4.4.4 Gestión de conceptos

La secuencia gestión de conceptos se divide en tres acciones elementales: listar conceptos, crear un concepto, edición de datos y borrado de conceptos.

1. *UIApplicationMain()*: El usuario abre la aplicación y este mensaje desencadena la activación de la aplicación por parte del sistema operativo.
  - 1.1. *applicationDidBecomeActive*: La aplicación está activa y se procede a la carga de la pantalla principal.
    - 1.1.1. *viewDidLoad*: La pantalla principal ya ha sido cargada y está lista para ser mostrada al usuario. Se muestra la pantalla al usuario y este ya puede interactuar con la aplicación.
2. *tableView:didSelectRowAtIndexPath*: El usuario ha seleccionado la opción de gestión de conceptos en la pantalla principal y ha desencadenado el evento de 'opción seleccionada'.
  - 2.1. La pantalla principal de la aplicación notifica a *UIKit* (el gestor de interfaz de usuario del SO) a través del método *performSegueWithIdentifier*: que ha sido seleccionada una opción de navegación hacia el gestor de conceptos y que debe instanciarlo y mostrarlo.
    - 2.1.1. *UIKit* ha instanciado el controlador de destino, este se ha cargado y justo en ese instante se ejecuta el método *viewWillAppear*: el cual indica que la vista asociada a este controlador está a punto de mostrarse. Este momento es la última oportunidad para recuperar los datos existentes para mostrarlos al usuario en cuanto aparezca la vista en pantalla.
      - 2.1.1.1. *SQConceptListController* invoca a su propio método *refreshData*: para cargar los datos que se van a listar en pantalla. El controlador se muestra en pantalla listo para que el usuario interactúe con él.
3. Para introducir nuevos datos el usuario debe acceder hasta la pantalla de listado de los mismos, una allí el usuario invoca la pantalla de adición/edición de sesiones a través de *UIKit*, pulsando en el botón correspondiente.
  - 3.1. *UIKit* invoca al método *presentViewController:animated*: que presenta al usuario el diálogo de adición/edición de conceptos. El usuario introduce los datos.
4. El usuario invoca desencadena el evento *saveButtonPressed*: pulsando en el botón de guardado de datos cuando ha finalizado
  - 4.1. El diálogo de adición/edición *SQConceptDialogController* el método *postNotificationName*: de *UIKit* para enviar la notificación de actualización a los consumidores de la información suscritos.
    - 4.1.1. *UIKit* invoca el método *refreshData*: de la clase *SQConceptListController*, suscrita a recibir notificaciones por parte del SO sobre cambios en la información que le interesa.
5. Para eliminar una sesión, el usuario pone en modo de edición la lista de sesiones a través de *setEditing:animated*:
6. El usuario le dice a *UIKit* que quiere borrar un elemento de la lista
  - 6.1. *UIKit* invoca a *commitEditingStyle:forRowAtIndexPath*: para que *SQConceptListController* efectúe el borrado de los datos.

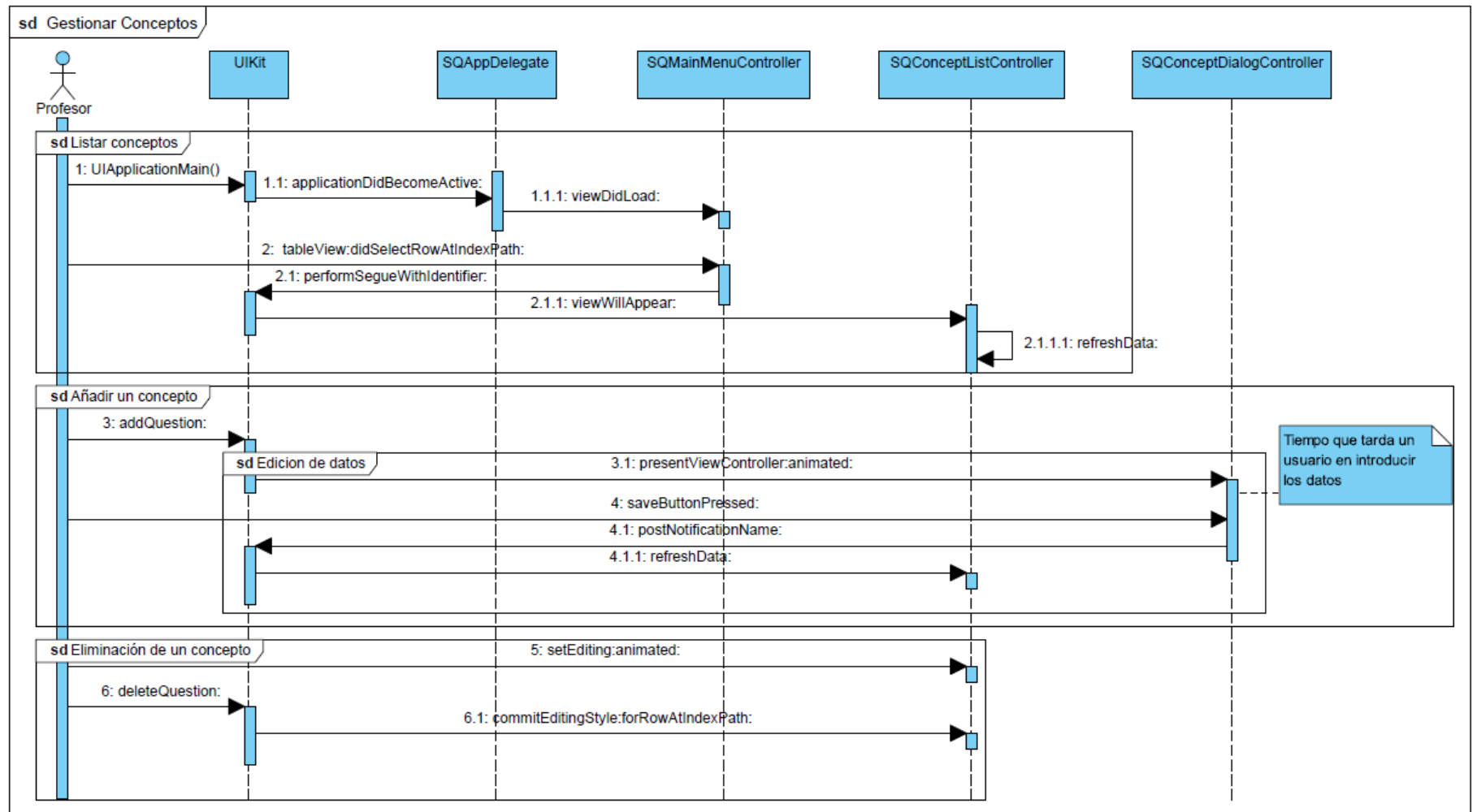


Ilustración 33: Diagrama de secuencia 'Gestionar Conceptos'

#### 4.4.5 Gestión de tweets

La secuencia gestión de tweets se divide en tres acciones elementales: listar tweets, crear un tweet y borrado de tweets.

1. *UIApplicationMain()*: El usuario abre la aplicación y este mensaje desencadena la activación de la aplicación por parte del sistema operativo.
  - 1.1. *applicationDidBecomeActive*: La aplicación está activa y se procede a la carga de la pantalla principal.
    - 1.1.1. *viewDidLoad*: La pantalla principal ya ha sido cargada y está lista para ser mostrada al usuario. Se muestra la pantalla al usuario y este ya puede interactuar con la aplicación.
2. *tableView:didSelectRowAtIndexPath*: El usuario ha seleccionado la opción de gestión de conceptos en la pantalla principal y ha desencadenado el evento de 'opción seleccionada'.
  - 2.1. La pantalla principal de la aplicación notifica a *UIKit* (el gestor de interfaz de usuario del SO) a través del método *performSegueWithIdentifier*: que ha sido seleccionada una opción de navegación hacia el gestor de conceptos y que debe instanciarlo y mostrarlo.
    - 2.1.1. *UIKit* ha instanciado el controlador de destino, este se ha cargado y justo en ese instante se ejecuta el método *viewWillAppear*: el cual indica que la vista asociada a este controlador está a punto de mostrarse. Este momento es la última oportunidad para recuperar los datos existentes para mostrarlos al usuario en cuanto aparezca la vista en pantalla.
      - 2.1.1.1. *SQSessionListController* invoca a su propio método *refreshData*: para cargar los datos que se van a listar en pantalla. El controlador se muestra en pantalla listo para que el usuario interactúe con él.
  3. *tableView:didSelectRowAtIndexPath*: El usuario ha seleccionado la opción de gestión de conceptos en la pantalla principal y ha desencadenado el evento de 'opción seleccionada'.
    - 3.1. La pantalla principal de la aplicación notifica a *UIKit* (el gestor de interfaz de usuario del SO) a través del método *performSegueWithIdentifier*: que ha sido seleccionada una opción de navegación hacia el gestor de conceptos y que debe instanciarlo y mostrarlo.
      - 3.1.1. *UIKit* ha instanciado el controlador de destino, este se ha cargado y justo en ese instante se ejecuta el método *viewWillAppear*: el cual indica que la vista asociada a este controlador está a punto de mostrarse. Este momento es la última oportunidad para recuperar los datos existentes para mostrarlos al usuario en cuanto aparezca la vista en pantalla.
        - 3.1.1.1. *SQSessionDetailsController* invoca a su propio método *refreshData*: para cargar los datos que se van a listar en pantalla. El controlador se muestra en pantalla listo para que el usuario interactúe con él.
  4. El usuario invoca añade un nuevo tweet a través de la acción *addTweet*:
    - 4.1. Una vez completada la inserción del nuevo tweet *SQSessionDetailsController* invoca a su propio método *refreshData*
  5. Para eliminar un tweet, el usuario pone en modo de edición la lista de tweets a través de *setEditing:animated*:
  6. El usuario le dice a *UIKit* que quiere borrar un elemento de la lista
    - 6.1. *UIKit* realiza una llamada al método *commitEditingStyle:forRowAtIndexPath*: de la clase *SQSessionDetailsController* para que efectúe el borrado de los datos.

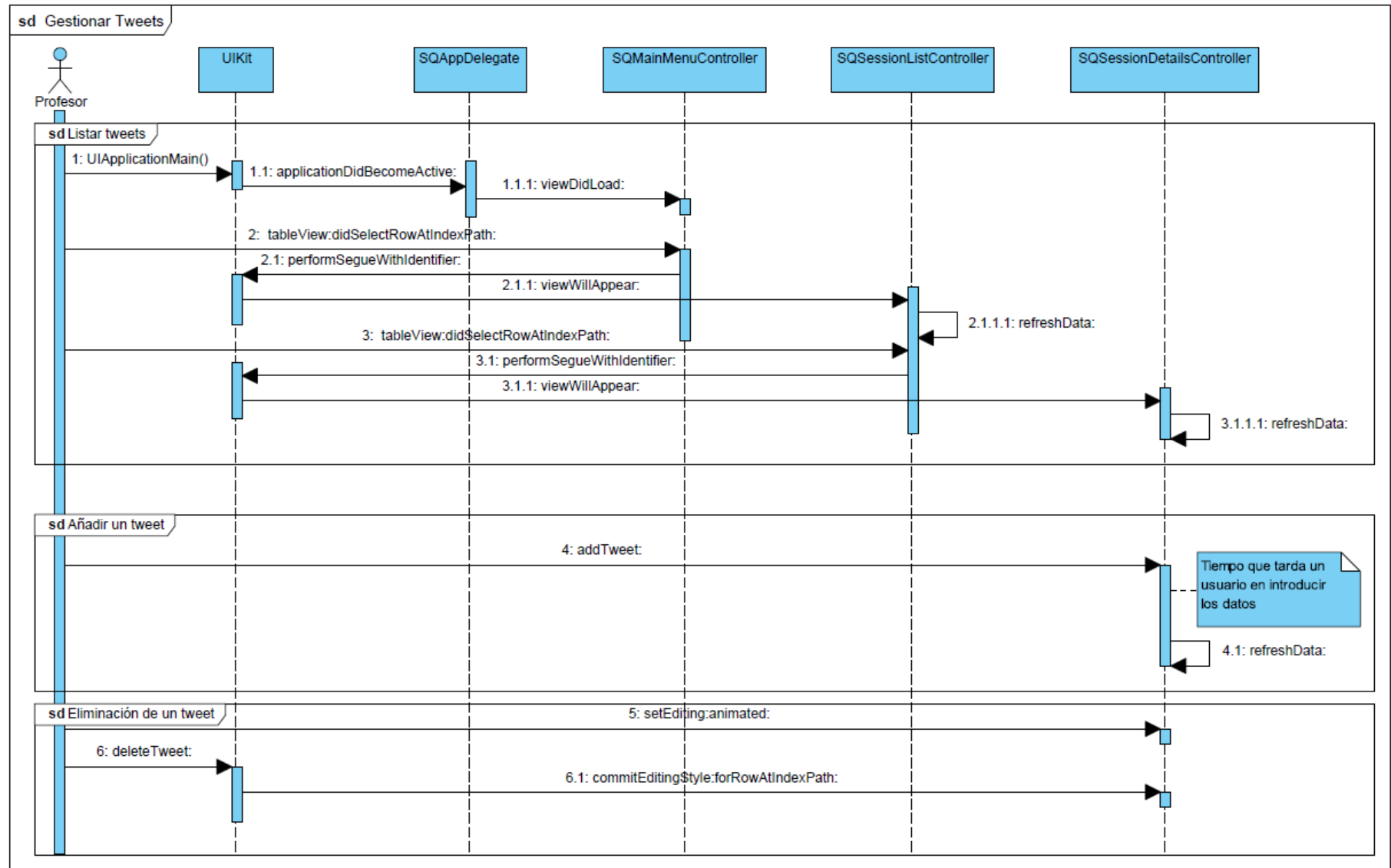


Ilustración 34: Diagrama de secuencia 'Gestionar Tweets'



#### 4.4.6 Consulta de estadísticas

La secuencia gestión de tweets es *per se* una acción elemental que se describe mediante la siguiente secuencia

1. *UIApplicationMain()*: El usuario abre la aplicación y este mensaje desencadena la activación de la aplicación por parte del sistema operativo.
  - 1.1. *applicationDidBecomeActive*: La aplicación está activa y se procede a la carga de la pantalla principal.
    - 1.1.1. *viewDidLoad*: La pantalla principal ya ha sido cargada y está lista para ser mostrada al usuario. Se muestra la pantalla al usuario y este ya puede interactuar con la aplicación.
2. *tableView:didSelectRowAtIndexPath*: El usuario ha seleccionado la opción de gestión de conceptos en la pantalla principal y ha desencadenado el evento de 'opción seleccionada'.
  - 2.1. La pantalla principal de la aplicación notifica a *UIKit* (el gestor de interfaz de usuario del SO) a través del método *performSegueWithIdentifier*: que ha sido seleccionada una opción de navegación hacia el gestor de conceptos y que debe instanciarlo y mostrarlo.
    - 2.1.1. *UIKit* ha instanciado el controlador de destino y su vista asociada, el controlador es notificado a través del método *viewDidLoad*:
      - 2.1.1.1. *SQStatsViewController* invoca al método *loadRequest*: del *WebView* que la clase maneja para cargar la página de estadísticas generada por el servidor bajo demanda.

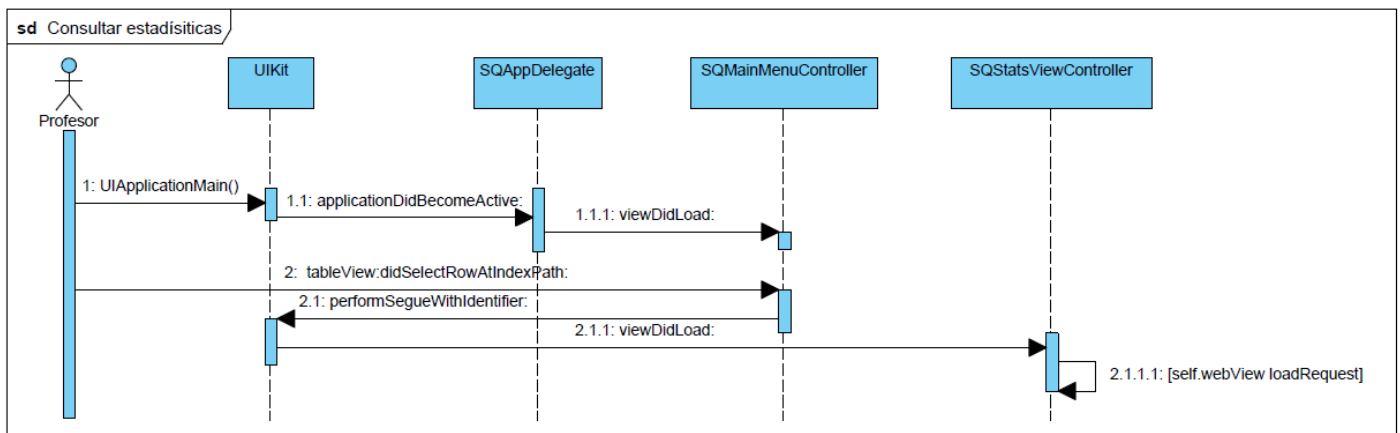


Ilustración 35: Diagrama de secuencia 'Consultar estadísticas'



---

## Capítulo 5: Pruebas del sistema

---

## 5.1 Especificación de pruebas del sistema

En el siguiente catálogo se describen todas las pruebas que forman parte del proyecto para conformar una especificación completa a partir de la cual testar la solución al problema planteado al inicio de este documento.

### 5.1.1 Formato de las pruebas

A continuación se describen los campos que definen cada una de las pruebas.

- **Identificador:** Código que identifica de manera única a la prueba. Tendrá el siguiente formato, **PX-YY** dónde **P** representa prueba, **X** el tipo de prueba que se define e **YY** es el identificador numérico de la prueba.
- **Descripción:** Descripción de la prueba, qué elemento o funcionalidad se va a probar.
- **Pasos:** Conjunto de pasos que deben seguirse para llevar a término la prueba.
- **Errores posibles:** Errores que pueden surgir durante la realización de la prueba
- **Requisitos:** Identificadores de los requisitos que quedan testados a través de la prueba.
- **Resultado esperado:** Resultado que se espera de la realización de la prueba. Si el resultado experimental coincide con el resultado documentado, la prueba ha sido exitosa.

Parámetro	Descripción
<b>Identificador</b>	PX-YYY
<b>Descripción</b>	Descripción de la prueba
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Primer paso</li> <li>2. Segundo paso</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• Posible error A</li> <li>• Posible error B</li> </ul>
<b>Requisitos</b>	RF-XXX, RNF-YYY
<b>Resultado esperado</b>	El resultado obtenido debe ser ABC

Tabla 75: Plantilla de ejemplo para la definición de pruebas

### 5.1.2 Catálogo de pruebas

A continuación se exponen todas las pruebas que la solución debe pasar para ser válida. El catálogo comprende un conjunto de pruebas de sistema que se realizan para validar todos y cada uno de los requisitos especificados<sup>6</sup> en la sección *Catálogo de requisitos* del presente documento.

Parámetro	Descripción
<b>Identificador</b>	PS-001
<b>Descripción</b>	Crear un curso
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción 'Cursos'</li> <li>2. En la lista de cursos pulsar el botón '+'</li> <li>3. Introducir los datos del curso y pulsar sobre 'Guardar'</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena el curso</li> </ul>
<b>Requisitos</b>	RF-001
<b>Resultado esperado</b>	En la lista de cursos debe aparecer el curso recién insertado

Tabla 76: Prueba PS-001 — Crear curso

<sup>6</sup> Dado que todas las pruebas van a ejecutarse en teléfonos y tabletas con sistema iOS 6.1 o superior y en algunas pruebas ha de hacerse uso del patrón *pull to refresh*, todas las pruebas de los requisitos no funcionales quedan implícitas dentro de las pruebas de este catálogo, salvo la del requisito RNF-005, la cual queda sujeta al proceso de certificación de aplicaciones que realiza Apple.

Parámetro	Descripción
<b>Identificador</b>	PS-002
<b>Descripción</b>	Modificación de un curso
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción ‘Cursos’</li> <li>2. En la lista de cursos pulsar el botón ‘Editar’</li> <li>3. Pulsar sobre el <i>disclosure indicator</i> azul que aparece junto al curso</li> <li>4. Modificar los datos que sean pertinentes</li> <li>5. Pulsar sobre el botón guardar</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena el curso modificado</li> </ul>
<b>Requisitos</b>	RF-002
<b>Resultado esperado</b>	En la lista de cursos debe aparecer el curso ya modificado

Tabla 77: Prueba PS-002 — Modificar curso

Parámetro	Descripción
<b>Identificador</b>	PS-003
<b>Descripción</b>	Borrado de un curso
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción ‘Cursos’</li> <li>2. En la lista de cursos pulsar el botón ‘Editar’</li> <li>3. Realizar la acción de borrado estándar de iOS</li> <li>4. Pulsar sobre el botón de eliminación</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se borra el curso</li> </ul>
<b>Requisitos</b>	RF-003
<b>Resultado esperado</b>	En la lista de cursos debe haber desaparecido el curso borrado y al forzar su actualización haciendo <i>pull to refresh</i> este no debe volver a aparecer en la lista

Tabla 78: Prueba PS-003 — Borrar curso

Parámetro	Descripción
<b>Identificador</b>	PS-004
<b>Descripción</b>	Crear una sesión
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción ‘Sesiones’</li> <li>2. En la lista de sesiones pulsar el botón ‘+’</li> <li>3. Introducir los datos de la sesión y pulsar sobre ‘Guardar’</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena la sesión</li> </ul>
<b>Requisitos</b>	RF-004
<b>Resultado esperado</b>	En la lista de sesiones debe aparecer la sesión recién insertada

Tabla 79: Prueba PS-004 — Crear sesión

Parámetro	Descripción
<b>Identificador</b>	PS-005
<b>Descripción</b>	Modificación de una sesión
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción ‘Sesiones’</li> <li>2. En la lista de sesiones pulsar el botón ‘Editar’</li> <li>3. Pulsar sobre el <i>disclosure indicator</i> azul que aparece junto a la sesión</li> <li>4. Modificar los datos que sean pertinentes</li> <li>5. Pulsar sobre el botón guardar</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena la sesión modificada</li> </ul>
<b>Requisitos</b>	RF-005
<b>Resultado esperado</b>	En la lista de sesiones debe aparecer la sesión ya modificada

Tabla 80: Prueba PS-005 — Modificación de una sesión

Parámetro	Descripción
<b>Identificador</b>	PS-006
<b>Descripción</b>	Borrado de una sesión
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción 'Sesión'</li> <li>2. En la lista de sesiones pulsar el botón 'Editar'</li> <li>3. Realizar la acción de borrado estándar de iOS</li> <li>4. Pulsar sobre el botón de eliminación</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se borra la sesión</li> </ul>
<b>Requisitos</b>	RF-006
<b>Resultado esperado</b>	En la lista de sesiones debe haber desaparecido la sesión borrada y al forzar su actualización haciendo <i>pull to refresh</i> esta no debe volver a aparecer en la lista

Tabla 81: Prueba PS-003 — Borrado de una sesión

Parámetro	Descripción
<b>Identificador</b>	PS-007
<b>Descripción</b>	Crear un concepto
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción 'Conceptos'</li> <li>2. En la lista de conceptos pulsar el botón '+'</li> <li>3. Introducir los datos del concepto y pulsar sobre 'Guardar'</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena el concepto</li> </ul>
<b>Requisitos</b>	RF-007
<b>Resultado esperado</b>	En la lista de cursos debe aparecer el concepto recién insertado

Tabla 82: Prueba PS-007 — Crear un concepto

Parámetro	Descripción
<b>Identificador</b>	PS-008
<b>Descripción</b>	Modificación de un concepto
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción 'Conceptos'</li> <li>2. En la lista de conceptos pulsar el botón 'Editar'</li> <li>3. Pulsar sobre el <i>disclosure indicator</i> azul que aparece junto al concepto</li> <li>4. Modificar los datos que sean pertinentes</li> <li>5. Pulsar sobre el botón guardar</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena el concepto modificado</li> </ul>
<b>Requisitos</b>	RF-008
<b>Resultado esperado</b>	En la lista de conceptos debe aparecer el concepto ya modificado

Tabla 83: Prueba PS-008 — Modificación de un concepto

Parámetro	Descripción
<b>Identificador</b>	PS-009
<b>Descripción</b>	Borrado de un concepto
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción 'Conceptos'</li> <li>2. En la lista de conceptos pulsar el botón 'Editar'</li> <li>3. Realizar la acción de borrado estándar de iOS</li> <li>4. Pulsar sobre el botón de eliminación</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se borra el concepto</li> </ul>
<b>Requisitos</b>	RF-009
<b>Resultado esperado</b>	En la lista de conceptos debe haber desaparecido el concepto borrado y al forzar su actualización haciendo <i>pull to refresh</i> este no debe volver a aparecer en la lista

Tabla 84: Prueba PS-009 — Borrado de un concepto

Parámetro	Descripción
<b>Identificador</b>	PS-010
<b>Descripción</b>	Crear una pregunta
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción 'Preguntas'</li> <li>2. En la lista de preguntas pulsar el botón '+'</li> <li>3. Introducir los datos de la sesión y pulsar sobre 'Guardar'</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena la pregunta</li> </ul>
<b>Requisitos</b>	RF-010
<b>Resultado esperado</b>	En la lista de preguntas debe aparecer la pregunta recién insertada

Tabla 85: Prueba PS-004 — Crear una pregunta

Parámetro	Descripción
<b>Identificador</b>	PS-011
<b>Descripción</b>	Modificación de una pregunta
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción 'Preguntas'</li> <li>2. En la lista de cursos pulsar el botón 'Editar'</li> <li>3. Pulsar sobre el <i>disclosure indicator</i> azul que aparece junto a la pregunta</li> <li>4. Modificar los datos que sean pertinentes</li> <li>5. Pulsar sobre el botón guardar</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena la pregunta modificada</li> </ul>
<b>Requisitos</b>	RF-011
<b>Resultado esperado</b>	En la lista de preguntas debe aparecer la pregunta ya modificada

Tabla 86: Prueba PS-011 — Modificación de una pregunta

Parámetro	Descripción
<b>Identificador</b>	PS-012
<b>Descripción</b>	Borrado de una pregunta
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción 'Preguntas'</li> <li>2. En la lista de cursos pulsar el botón 'Editar'</li> <li>3. Realizar la acción de borrado estándar de iOS</li> <li>4. Pulsar sobre el botón de eliminación</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se borra la pregunta</li> </ul>
<b>Requisitos</b>	RF-012
<b>Resultado esperado</b>	En la lista de preguntas debe haber desaparecido la pregunta borrada y al forzar su actualización haciendo <i>pull to refresh</i> esta no debe volver a aparecer en la lista

Tabla 87: Prueba PS-012 — Borrado de una pregunta

Parámetro	Descripción
<b>Identificador</b>	PS-013
<b>Descripción</b>	Crear un tweet
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción ‘Sesiones’</li> <li>2. En la lista de sesiones seleccionar la sesión en la que se quiere almacenar un tweet</li> <li>3. Seleccionar la sección ‘Tweets’</li> <li>4. Pulsar sobre el botón ‘+’</li> <li>5. Introducir los datos del tweet</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena el tweet</li> </ul>
<b>Requisitos</b>	RF-013
<b>Resultado esperado</b>	En la lista de tweets dentro de una sesión debe aparecer el curso tweet recién insertado

Tabla 88: Prueba PS-013 — Crear un tweet

Parámetro	Descripción
<b>Identificador</b>	PS-014
<b>Descripción</b>	Borrar un tweet
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción ‘Sesiones’</li> <li>2. En la lista de sesiones seleccionar la sesión en la que se quiere almacenar un tweet</li> <li>3. Seleccionar la sección ‘Tweets’</li> <li>4. Pulsar sobre el botón ‘+’</li> <li>5. Introducir los datos del tweet</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se almacena el tweet</li> </ul>
<b>Requisitos</b>	RF-014
<b>Resultado esperado</b>	En la lista tweets dentro de una sesión debe haber desaparecido el tweet borrado y al forzar su actualización haciendo <i>pull to refresh</i> este no debe volver a aparecer en la lista

Tabla 89: Prueba PS-014 — Borrar un tweet

Parámetro	Descripción
<b>Identificador</b>	PS-015
<b>Descripción</b>	Consulta de estadísticas
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Seleccionar en el menú principal la opción ‘Resultados’</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• La pantalla queda en blanco porque no pueden recuperarse las estadísticas</li> </ul>
<b>Requisitos</b>	RF-015
<b>Resultado esperado</b>	Debe aparecer la pantalla de resultados con todas aquellas preguntas que se hayan lanzado y respondido

Tabla 90: Prueba PS-015 — Consulta de estadísticas



Parámetro	Descripción
<b>Identificador</b>	PS-016
<b>Descripción</b>	Sincronización
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Vincular la aplicación con Dropbox a través de la opción 'Enlazar con Dropbox' del menú principal en todos los dispositivos a probar. Asegurarse de que existe conexión a Internet</li> <li>2. Añadir, borrar o modificar los datos que se deseen</li> <li>3. Abrir la aplicación en los demás dispositivos</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• Los datos no se sincronizan y difieren entre dispositivos</li> </ul>
<b>Requisitos</b>	RF-016
<b>Resultado esperado</b>	En el resto de dispositivos la aplicación debe abrirse con los datos en el mismo estado que los del dispositivo que hizo la última modificación.

Tabla 91: Prueba PS-016 — Sincronización

Parámetro	Descripción
<b>Identificador</b>	PS-017
<b>Descripción</b>	<i>Check-in</i> con Twitter
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se debe abrir la aplicación cliente, seleccionar una cuenta twitter de las disponibles en el dispositivo y pulsar el botón de inicio de sesión</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• La aplicación se queda trabada en un indicador de progreso indeterminado</li> </ul>
<b>Requisitos</b>	RF-017
<b>Resultado esperado</b>	La aplicación debe mostrar un indicador de progreso indeterminado y después debe pasar a la pantalla principal

Tabla 92: Prueba PS-017 — *Check-in* con Twitter

Parámetro	Descripción
<b>Identificador</b>	PS-018
<b>Descripción</b>	Suscribirse a una sesión
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se debe abrir la aplicación cliente y hacer <i>check-in</i> en el sistema</li> <li>2. En la lista de sesiones disponibles (de haberlas, si no debe crearse previamente una con fines de prueba) desplazar el interruptor de suscripción de una de ellas</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se puede mostrar la lista de sesiones porque no hay conexión</li> <li>• No existen sesiones disponibles</li> <li>• No se puede completar el proceso y el interruptor vuelve a su posición original</li> </ul>
<b>Requisitos</b>	RF-018
<b>Resultado esperado</b>	El interruptor de suscripción a una sesión debe desplazarse y mantenerse en la posición de activado para confirmar que se está suscrito a una sesión.

Tabla 93: Prueba PS-018 — Suscribirse a una sesión

Parámetro	Descripción
<b>Identificador</b>	PS-019
<b>Descripción</b>	Recibir preguntas
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se debe abrir la aplicación cliente y hacer <i>check-in</i> en el sistema</li> <li>2. Suscribirse a una sesión. Si no existe, crearla.</li> <li>3. Lanzar una pregunta a dicha sesión</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se puede recibir la pregunta por no haber conexión a internet o un fallo en el servidor de envío de notificaciones</li> </ul>
<b>Requisitos</b>	RF-019
<b>Resultado esperado</b>	Si la aplicación está abierta, la pregunta aparece inmediatamente en pantalla. Si la aplicación está cerrada o en segundo plano, aparece una alerta que pulsándola coloca la pregunta en pantalla.

Tabla 94: Prueba PS-019 — Recibir preguntas

Parámetro	Descripción
<b>Identificador</b>	PS-020
<b>Descripción</b>	Responder preguntas
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Se debe abrir la aplicación cliente y hacer <i>check-in</i> en el sistema</li> <li>2. Suscribirse a una sesión. Si no existe, crearla.</li> <li>3. Lanzar una pregunta a dicha sesión</li> <li>4. Enviar una respuesta</li> </ol>
<b>Posibles errores</b>	<ul style="list-style-type: none"> <li>• No se puede recibir la pregunta por no haber conexión a internet o un fallo en el servidor de envío de notificaciones</li> </ul>
<b>Requisitos</b>	RF-020
<b>Resultado esperado</b>	Si la respuesta se ha enviado correctamente, en la pantalla de resultados debería verse la pregunta, la respuesta seleccionada y en la lista de usuarios que han seleccionado dicha respuesta, el usuario empleado para hacer la prueba. También debería mostrarse como leída (marcada con un punto amarillo) en la lista de preguntas recibidas.

Tabla 95: Prueba PS-020 — Responder preguntas

### 5.1.3 Trazabilidad de pruebas de sistema – requisitos funcionales

En esta sección se muestra la correspondencia entre pruebas de sistema y requisitos funcionales.

	RF-001	RF-002	RF-003	RF-004	RF-005	RF-006	RF-007	RF-008	RF-009	RF-010	RF-011	RF-012	RF-013	RF-014	RF-015	RF-016	RF-017	RF-018
PS-001	X																	
PS-002		X																
PS-003			X															
PS-004				X														
PS-005					X													
PS-006						X												
PS-007							X											
PS-008								X										
PS-009									X									
PS-010										X								
PS-011											X							
PS-012												X						
PS-013													X					
PS-014														X				
PS-015															X			
PS-016																X		
PS-017																	X	
PS-018																		X

Tabla 96: Matriz de trazabilidad pruebas de sistema – requisitos funcionales



---

## Capítulo 6: Presupuesto

---

## 6.1 Presupuesto del proyecto

En esta sección del documento se va a proceder al cálculo final y real de los costes asociados al desarrollo del proyecto, especificando los costes reales de personal y materiales empleados durante el proceso de desarrollo.

### 6.1.1 Costes de personal

En este apartado se especifican los costes del personal involucrado durante la elaboración del proyecto. El personal incluido consta de Roberto Estrada Casarrubios, desarrollador principal del proyecto, a Alberto Heredia García y Javier García Guzmán, tutor y director del proyecto respectivamente. También se incluye a José Ángel Cuadrado Mingo debido a las horas de ingeniería que ha debido de invertir para la integración de su proyecto con este.

En la siguiente tabla se desglosa el coste de cada miembro del equipo

Integrante	Categoría	Coste hombre/hora (€)	Horas totales	Coste (€)
Roberto Estrada Casarrubios	Ingeniero Junior	18	769,5	13.851
Javier García Guzmán	Ingeniero Senior	28	18	504
Alberto Heredia García	Ingeniero Senior	28	18	504
José A. Cuadrado Mingo	Ingeniero Junior	18	18	324

Tabla 97: Costes de personal

El coste total del personal es la suma de todos los costes individuales de cada uno de los miembros del equipo, por lo tanto el coste total de personal asciende a **15.183 €**

### 6.1.2 Costes de materiales

Se deben tener en cuenta como gastos asociados a la realización del proyecto las amortizaciones de los materiales empleados.

Se ha considerado un tiempo de vida útil para los ordenadores de 48 meses, mientras que para los dispositivos móviles se ha considerado una vida útil de 18 meses, que es lo que suele tardar desde el lanzamiento de un dispositivo hasta que aparece la siguiente generación del mismo.

La licencia de desarrollador es un coste fijo anual.

El coste imputable se ha calculado utilizando los siguientes datos:

- A: número de meses desde la fecha de facturación en que el equipo es utilizado
- B: periodo de amortización (48 o 18 meses según tipo de dispositivo)
- C: coste del equipo (sin IVA)
- D: Porcentaje de uso que se dedica al proyecto

Teniendo en cuenta los anteriores datos, el coste imputable se calcula mediante la siguiente fórmula de cálculo de amortización:

$$A/B * C * D$$

Material	Meses de uso	Meses amortización	Precio (€)	% dedicación al proyecto	Coste para el proyecto (€)
iMac 21.5'	8,5	48	1.145	100%	202,76
iPad 2	8,5	18	629	20%	59,40
iPad Mini	8,5	18	329	20%	31,07
Portátil ASUS A52J	4	48	520	100%	43,33
Windows 8 Pro	4	48	199	100%	16,58
Office Hogar y Empresas 2013	2	24	269	100%	22,41
Licencia desarrollador Apple	8,5	12	80	100%	56,66

Tabla 98: Costes de materiales

El coste total del materiales es la suma de todos los costes para el proyecto de cada uno de los materiales, por lo tanto el coste total de materiales asciende a **432,21 €**

### 6.1.3 Otros costes

En este apartado se incluirán otros costes que no pueden enmarcarse en costes de personal o de materiales, tales como costes de desplazamiento, material fungible, etc.

Concepto	Meses de uso	Precio (€/mes)	% dedicación al proyecto	Coste para el proyecto (€)
Abono de transportes zona B1	9	63,70	75%	429,97

Tabla 99: Otros costes

### 6.1.4 Coste total

El coste total del proyecto es la suma de todos los costes parciales especificados en cada una de las categorías anteriormente desglosadas por tanto, se tiene que:

Categoría	Valor
Costes de personal	15.183 €
Costes de materiales	432,21 €
Otros costes	429,97 €
<b>TOTAL</b>	<b>16.045,18 €</b>

Tabla 100: Coste total del proyecto

Se concluye que el coste final del desarrollo del proyecto asciende a **16.045,18 € (dieciséis mil cuarenta y cinco euros con diecisiete siete céntimos)**.





---

## Capítulo 7: Conclusiones y líneas futuras

---

## 7.1 Conclusiones

El proyecto de fin de carrera es el último paso de una larga carrera de fondo a cuyo final tan sólo unos pocos llegan y que debemos afrontar todos los futuros ingenieros para demostrar nuestras aptitudes en el ejercicio de una exigente profesión en la que siempre se espera que demos lo mejor de nosotros mismos como profesionales y como personas, allá donde la sociedad nos necesite.

Muchos ven en la realización del proyecto de fin de carrera un trámite —algo que hay que quitarse de encima cuanto más rápido, mejor— para obtener un título, un mero billete al mundo laboral y no como la forma de ganarse el respeto de quienes ejercen la misma profesión que desempeñaron personas de la talla de Arquímedes, Leonardo, los Hermanos Wright o en el campo de la computación Ada Lovelace, Alan Turing, John von Neumann o Dennis Ritchie entre muchos otros.

En ese espíritu este proyecto ha sido tomado no como un escollo sino como un reto personal donde poner en juego muchos de los conocimientos adquiridos a lo largo de todos estos años y de aprender multitud de cosas nuevas debido a la gran libertad para escoger y probar las tecnologías más modernas, incluso experimentales, existentes en el momento dentro del mundo del desarrollo software y en especial del software para dispositivos móviles. Una oportunidad que no se tiene frecuentemente.

Esta libertad no obstante tiene un precio en tiempo y recursos y a veces esto ha salido bien ahorrando mucho, como con el uso de *Dropbox Datastore* para el desarrollo del almacén de datos en la nube y otras ha salido mal, como cuando se decidió usar *Ruby On Rails* para implementar el *backend* del sistema y hubo que reescribirlo usando *Java* y *Play! Framework* (otro acierto) o cuando se empleó *iCloud*, el cual fallaba estrepitosamente y ha traído muchos disgustos a otros desarrolladores que apostaron por él y obtuvieron malas valoraciones de sus *apps* en la tienda de aplicaciones móviles de *Apple*.

Este proyecto está encuadrado en una línea de investigación sobre aplicación de tecnologías móviles a la educación dentro del grupo de investigación miSEL del *Software Engineering Lab* de la Universidad Carlos III de Madrid. Embarcarse en un proyecto que va a formar parte de un sistema completo que servirá para ayudar a poner en práctica nuevas técnicas y recursos docentes para la mejora de la educación, pilar fundamental de esta sociedad, es una tarea gratificante, una gran responsabilidad y una oportunidad única puesto que el desarrollo para la plataforma iOS no es precisamente barato como se ha visto en la sección *Costes de materiales* del presupuesto, costes que para un estudiante son aún más difíciles de asumir.

Haber invertido algo más de tiempo a cambio de los conocimientos y experiencia adquirida bien ha merecido la pena, pues, con el auge y omnipresencia de los dispositivos y aplicaciones móviles toda clase de experiencia en este nicho de mercado constituye una gran ventaja competitiva de cara a un mercado laboral que cada día demanda más profesionales versados en este tipo de tecnologías, sobre todo en la plataforma iOS.

La plataforma iOS, aunque ya no es la más extendida a nivel de dispositivos sí sigue siendo la que más volumen de negocio posee debido a que sus usuarios emplean más *apps* de pago que los del resto de plataformas, lo cual obliga a desarrollar aplicaciones de una gran calidad por tres motivos:

- Para pasar el riguroso proceso de certificación y control de calidad al que *Apple* somete a las aplicaciones antes de publicarlas en la tienda.
- Porque la competencia es feroz y la única manera de sobresalir en este mercado saturado es destacando sobre los rivales.
- Porque el usuario está acostumbrado a pagar por la mayoría de *apps* y entiende dicho pago como una garantía de un estándar de calidad. Si se ofrece un producto de baja calidad, que no esté a la altura de las expectativas de los usuarios, todo habrá acabado antes de empezar.

## 7.2 Líneas futuras

Todo lo visto durante este documento no son más que los cimientos de un prototipo de un producto que aspira a ser adoptado por profesores y alumnos como parte del proceso de transformación que sufrirá en el futuro a corto plazo la educación a través del uso de dispositivos móviles, cada día más presentes en las aulas.

### 7.2.1 Adaptación a iOS 7

Debido al reciente lanzamiento de la séptima gran versión del sistema operativo iOS y el radical cambio en el diseño de la interfaz de usuario que introduce se decidió dejar para el futuro la adaptación a dicha versión del SO. No obstante, mientras se escribían estas líneas el desarrollo ha continuado y a continuación se muestra una captura de pantalla de una versión preliminar adaptada para iOS 7.



Ilustración 37: Imagen preliminar de la versión iOS 7

### 7.2.2 Subida de ficheros, videos y otros recursos

Hasta el momento, los únicos recursos que pueden asociarse a las sesiones de clase son tweets que los alumnos pueden leer a través de la cuenta de la cuenta twitter asociada a la asignatura, o preguntas que el profesor puede lanzar a sus alumnos y que estos responderán desde sus dispositivos.

En futuras versiones, se incluirá un apartado de subida de ficheros, desde donde el profesor podrá colgar ficheros como por ejemplo, presentaciones de diapositivas que luego podrá cargar en la aplicación de impartición de clases mTeachClass simplemente seleccionando la sesión que quiere impartir.

Este proceso de subida de ficheros actualmente es manual y tedioso, con lo cual el profesor ganaría mucha comodidad a la hora de preparar sus clases.

También el profesor —aprovechando la integración existente— puede seleccionar qué ficheros de su cuenta de Dropbox quiere hacer disponibles al público.

En cuanto a los vídeos, el profesor también podrá seleccionar qué videos quiere proyectar durante la clase y llegado el momento de dar la clase estos quedarán automáticamente disponibles para mTeachClass y podrán ser proyectados sin hacer absolutamente nada más.

### 7.2.3 Mejora del sistema de preguntas

Actualmente las preguntas sólo soportan texto plano y no tienen un tiempo limitado de respuesta salvo que se finalice la sesión, en cuyo caso nadie más podrá responder una pregunta.

De cara al futuro sería positivo poder incluir ilustraciones o contenido enriquecido con HTML5 dentro de los enunciados para soportar la realización de preguntas mucho más complejas.

### 7.2.4 Integración con otros sistemas LMS

Integrar con otros sistemas LMS proporcionaría una gran flexibilidad a los docentes de usar el sistema que empleen sus instituciones a modo de campus virtual. En la Universidad Carlos III se emplea Moodle, uno de los más extendidos en la comunidad educativa aunque también destacan otros sistemas como Chamilo.

La integración con estos sistemas proporcionaría una mayor comodidad a los profesores, quienes podrían usar una herramienta u otra indistintamente, según les venga mejor.

### 7.2.5 Integración con otras redes sociales

En esta versión se ha hecho uso de la red de Twitter pero podrían emplearse otras como Facebook o Google+ para gestionar grupos o comunidades, gestionar un chat de clase, realizar publicaciones más extensas o simplemente para que los estudiantes puedan identificarse mediante el servicio de la red social que prefieran o de la que dispongan cuenta. En este último caso, Google+ debería tener prioridad debido a la integración del correo de la Universidad Carlos III con el servicio de Google Apps, ofreciendo un servicio de *single sign-on* a los estudiantes a través de Google Apps.

### 7.2.6 Mejoras de *backend*

El *backend* de la aplicación deberá ser ampliado para acoger servicios como notificaciones *push* a dispositivos Android, securización de la API a través del uso del estándar OAuth, optimización para ser escalable a muchos usuarios, etc.

---

# Referencias

---

- [1] I. Sørensen, «ON STUDENT RESPONSE SYSTEM – SRS,» *Proceedings of EDULEARN13 Conference*, 2013.
- [2] Educause Learning Initiative, «7 things you should know about Personal Learning Environments,» 2009. [En línea]. Available: <https://net.educause.edu/ir/library/pdf/ELI7049.pdf>. [Último acceso: 14 Octubre 2013].
- [3] EduClick S.L., «Mandos de Respuesta: EduClick,» [En línea]. Available: <http://www.educlick.es/mandos.html>. [Último acceso: 15 Octubre 2013].
- [4] T. . M. Thorseth, G. Hansen-Nygård, J. B. Stav, K. Bjørkli y P. Pein, «SRS overview,» 17 Abril 2012. [En línea]. Available: [http://www.histproject.no/sites/histproject.no/files/SRS\\_overview\\_newest.pdf](http://www.histproject.no/sites/histproject.no/files/SRS_overview_newest.pdf). [Último acceso: 15 Octubre 2013].
- [5] Sør-Trøndelag University College, «Products | histproject.no,» [En línea]. Available: <http://www.histproject.no/node/7>. [Último acceso: 15 Octubre 2013].
- [6] Nielsen, «Nielsen tops of 2012: Digital,» 20 Diciembre 2012. [En línea]. Available: <http://www.nielsen.com/us/en/newswire/2012/nielsen-tops-of-2012-digital.html>. [Último acceso: 16 Octubre 2013].
- [7] J. A. Cuadrado Mingo, Plataforma para el uso de Twitter en la preparación e impartición de clases presenciales, Leganés, 2013.
- [8] Apple Inc., «iOS Technology Overview,» 19 Septiembre 2012. [En línea]. Available: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechOverview.pdf>. [Último acceso: 17 Agosto 2013].
- [9] Apple Inc., «Object-Oriented Programming with Objective-C,» 15 Noviembre 2010. [En línea]. Available: [https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/OOP\\_ObjC/OOP\\_ObjC.pdf](https://developer.apple.com/library/ios/documentation/Cocoa/Conceptual/OOP_ObjC/OOP_ObjC.pdf). [Último acceso: 18 Agosto 2013].
- [10] Wikipedia contributors, «Xcode,» 14 Agosto 2013. [En línea]. Available: <http://en.wikipedia.org/w/index.php?title=Xcode&oldid=568488413>. [Último acceso: 19 Agosto 2013].
- [11] Apple Inc., «Developer Tools Features,» 28 Enero 2013. [En línea]. Available: <https://developer.apple.com/technologies/tools/features.html>. [Último acceso: 19 Agosto 2013].
- [12] E. Duran y F. Pelosin, «CocoaPods Docs,» 11 Noviembre 2011. [En línea]. Available: <http://docs.cocoapods.org/cocoapods/index.html>. [Último acceso: 20 Agosto 2013].
- [13] R. Esteban Santiago, A. Heredia García, J. García Guzmán y A. de Amescua, «Mass-Market Application Development Using Agile Techniques: How Agile Are We Really?,» de *Systems, Software and Services Process Improvement*, Springer Berlin Heidelberg, 2013, pp. 259-269.
- [14] K. Schwaber and J. Sutherland, *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*, Lexington, Massachusetts: scrum.org, 2013.
- [15] D. Peterson, "What is Kanban?," 1 Enero 2009. [Online]. Available: <http://www.kanbanblog.com/explained/index.html>. [Accessed 28 Agosto 2013].
- [16] J. Little, «Why I prefer ScrumBan to Kanban,» 1 Junio 2013. [En línea]. Available: <http://www.leanagiletraining.com/blog/key-problems/why-i-prefer-scrumban/>. [Último acceso: 29 Agosto 2013].
- [17] R. Esteban Santiago, Evaluación y mejora de procesos software para equipos de desarrollo ágil, Leganés: Universidad Carlos III de Madrid, 2013.
- [18] The Institute of Electrical and Electronics Engineers, Inc, *IEEE Recommended Practice for Software Requirements Specifications*, New York, 1998.
- [19] Apple Inc., "iOS Human Interface Guidelines," Cupertino, 2013.
- [20] M. Cohn, "User Stories," 2012. [Online]. Available: <http://www.mountaingoatsoftware.com/topics/user-stories>. [Accessed 4 Septiembre 2013].
- [21] M. Cohn, «Advantages of the “As a user, I want” user story template,» 25 Abril 2008. [En línea]. Available: <http://www.mountaingoatsoftware.com/blog/advantages-of-the-as-a-user-i-want-user-story-template>. [Último acceso: 4 Septiembre 2013].
- [22] Dropbox, "Dropbox - Datastore API Tutorial," 1 Agosto 2013. [Online]. Available: <https://www.dropbox.com/developers/datastore/tutorial/ios>.
- [23] Apple Inc., «Cocoa Core Competencies,» 8 Agosto 2013. [En línea]. Available: <http://developer.apple.com/library/ios/documentation/general/conceptual/devpedia-cocoacore/MemoryManagement.html>. [Último acceso: 19 Agosto 2013].
- [24] D. Almena García, Aplicación móvil basada en redes sociales para dar soporte a la impartición de clases (Plataforma Android), Leganés: Universidad Carlos III de Madrid, 2013.

- [25] L. J. Sanz Díaz, Aplicación móvil basada en redes sociales para dar soporte a la realización de exámenes tipo test (HTML 5), Leganés: Universidad Carlos III de Madrid, 2013.
- [26] L. Sanz González, Aplicación móvil basada en redes sociales para dar soporte a la realización de exámenes tipo test (Plataforma Android), Leganés: Universidad Carlos III de Madrid.

---

## **Apéndice A: Manual de usuario**

---

## A.1 mTeachPlan: Manual del profesor

mTeachPlan para iOS es una aplicación hecha para profesores que le ayudará en su día a día como profesor a preparar los recursos y materiales de sus clases por adelantado, en cualquier momento y lugar, a través de un dispositivo iOS como una tableta iPad o un teléfono iPhone para después impartirlas mediante mTeachClass.

### A.1.1 Visión general de mTeachPlan

mTeachPlan le permite gestionar múltiples asignaturas, dentro de las cuales puede planificar múltiples sesiones de clase, qué conceptos se van a impartir durante la sesión y asociar materiales docentes a dichos conceptos —por el momento sólo tweets y preguntas de clase— que se usarán posteriormente a la hora de impartir las clases.

Nada más abrir la aplicación podrá ver la vista principal, donde podrá acceder a todas las opciones de la aplicación.



Ilustración 38: Menú principal de mTeachPlan

Aunque mTeachPlan se puede usar por sí misma a modo de base de datos para anotar los datos que necesite para planificar sus clases su verdadera potencia se descubre cuando coopera con el resto de aplicaciones de la suite mTeach. Para ello se debe disponer de una cuenta de usuario en Dropbox.com, un servicio de almacenamiento de datos en la nube que mantendrá sincronizados los datos entre todas las aplicaciones mTeach.

Si cambia de dispositivo o lo pierde, sus datos estarán a salvo. Las aplicaciones mTeach cubren sus espaldas.



## A.1.2 Sincronización

Para que las aplicaciones mTeach cooperen tan sólo debe pulsar la primera vez que use la aplicación sobre el botón *Enlazar con Dropbox* y aparecerá la siguiente pantalla.



Ilustración 39: Enlace de la aplicación con Dropbox

Introduzca su usuario y contraseña de Dropbox en la pantalla y pulse sobre *Iniciar Sesión* y la aplicación estará lista para compartir datos.

Si ya tiene instalada en su dispositivo la aplicación oficial de Dropbox, simplemente se abrirá solicitándole los permisos correspondientes, una vez autorizado volverá a la pantalla de mTeachPlan.

## A.1.3 Planificación de cursos

Para planificar un nuevo curso, tan solo hay que pulsar en la opción *Cursos* del menú principal y luego en el botón con el signo “+” para añadir un curso nuevo y rellenar los datos asociados, o en el botón *Editar* para modificar o borrar los ya existentes.



Ilustración 40: Pantalla de creación de cursos



Ilustración 41: Pantalla de edición de cursos

### A.1.4 Planificación de sesiones

Para planificar sesiones, tan solo hay que pulsar en la opción *Sesiones* del menú principal y luego en el botón con el signo “+” en la parte inferior de la pantalla de lista de sesiones para añadir una nueva y rellenar los datos asociados, o en el botón *Editar* para modificar o borrar los ya existentes.



Ilustración 42: Pantalla de creación de sesiones

Las sesiones deben asociarse a un curso para luego poder impartirlas desde mTeachClass aunque pueden crearse sin asociar a ningún curso y luego más tarde asignarlo modificando la sesión desde la pantalla de edición.

Una vez creada una sesión, se puede acceder a sus contenidos pulsando sobre el título de la sesión que se desee en la lista de sesiones.

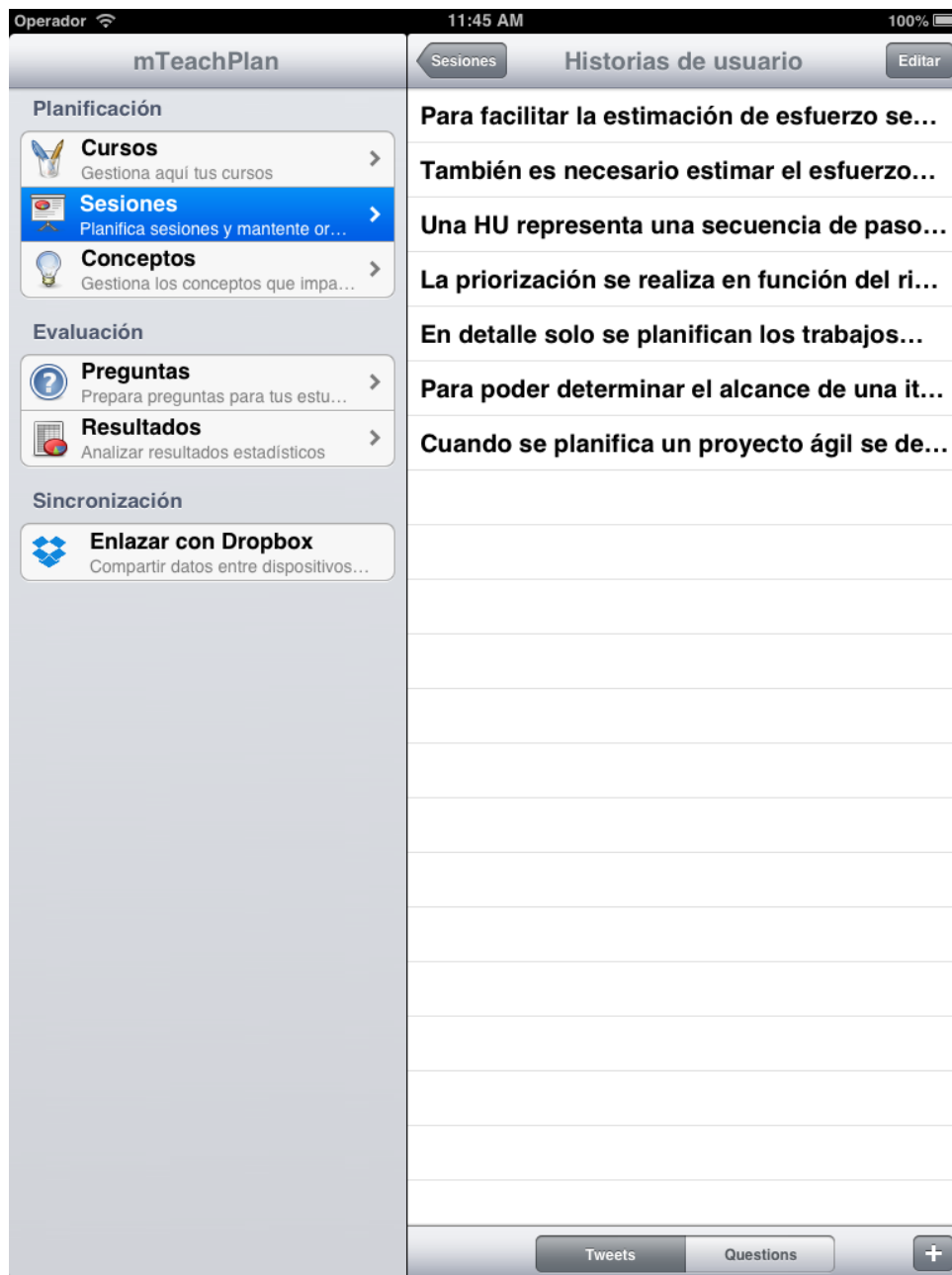


Ilustración 43: Contenidos de una sesión

Desde esta pantalla se pueden añadir recursos a una sesión simplemente pulsando en la parte inferior el tipo de recurso a añadir —actualmente tweets y preguntas— y pulsando el botón con el signo “+” aparecerá la pantalla de creación correspondiente.

### A.1.5 Gestión de preguntas y conceptos

La gestión de preguntas y conceptos funciona de forma análoga a lo visto en el apartado *Planificación de sesiones*. Simplemente hay que acudir en el menú principal a sus respectivas secciones. En el caso de las preguntas, estas también pueden gestionarse como se ha visto en el apartado *Planificación de sesiones* desde dentro de los detalles de una sesión.



Ilustración 44: Pantalla de creación de conceptos.

### A.1.6 Resultados

En la sección de resultados se mostrarán al profesor las preguntas que ha lanzado durante una clase y podrá ver en detalle qué porcentaje de alumnos han respondido correctamente, qué porcentaje de alumnos han seleccionado qué opción y finalmente, dentro de cada opción podrá ver quienes han sido los alumnos que han respondido dicha opción.

## A.2 mClassTest: Manual del alumno

mClassTest es una aplicación pensada para que los estudiantes puedan responder en tiempo real durante la clase a las preguntas que les formulen sus profesores durante la misma. Lo único que deben hacer es instalarla en sus dispositivos, iniciar sesión empleando una cuenta Twitter, suscribirse a la sesión que haya creado el profesor y esperar a recibir preguntas.

### A.2.1 Iniciar sesión

Empezar a usar mClassTest es muy simple, tan sólo hay que abrir la aplicación y seleccionar una cuenta de twitter configurada en el dispositivo. Si no hay ninguna configurada, deberá acudir a la aplicación de ajustes del sistema operativo y configurar una.



Ilustración 45: Pantalla de inicio de sesión de mClassTest

### A.2.2 Suscribirse a una sesión

Una vez iniciada sesión es muy fácil suscribirse a una sesión, tan solo hay que marcar las sesiones a las que se desea estar suscrito en la lista de sesiones disponibles que aparecerá.

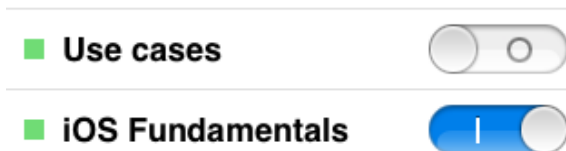


Ilustración 46: Ejemplo de suscripción a sesiones

También un alumno puede suscribirse con la sesión ya iniciada y en ese caso es probable que el profesor ya haya publicado algunas preguntas, para ello se ha creado la bandeja de entrada de preguntas, en la cual el alumno haciendo un gesto *pull to refresh* puede obtener las preguntas que hay pendientes para él.

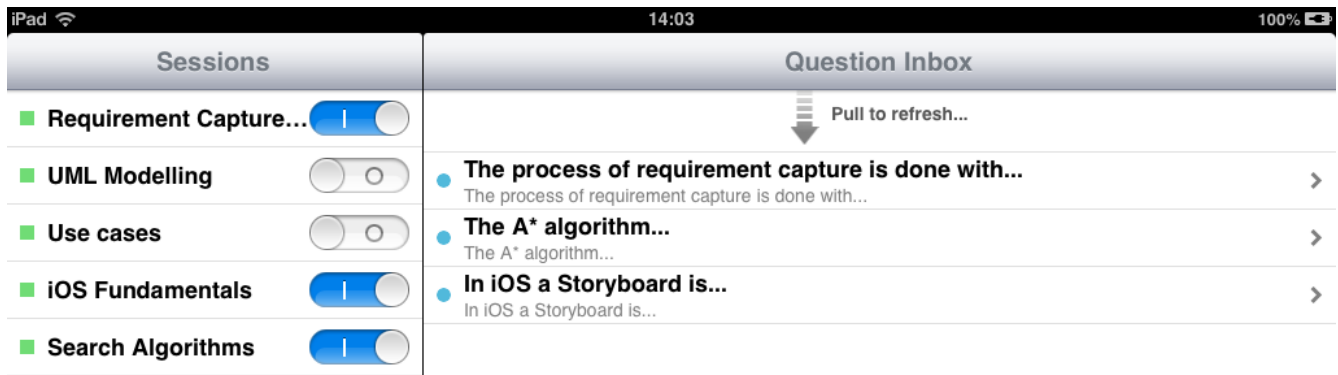


Ilustración 47: Sesiones suscritas y bandeja de entrada de preguntas.

### A.2.3 Responder preguntas

Los alumnos pueden responder preguntas mientras las sesiones a las que están suscritos estén abiertas, para ello tienen tres caminos.

- Seleccionar una pregunta no respondida en la bandeja entrada (punto azul) o modificar una respuesta a una pregunta ya respondida (punto amarillo)
- Tener la aplicación en funcionamiento y cuando un profesor lance una pregunta inmediatamente se mostrará la pantalla de respuesta de preguntas
- Recibir una notificación en la que pulsando se mostrará la pantalla de responder a la pregunta.

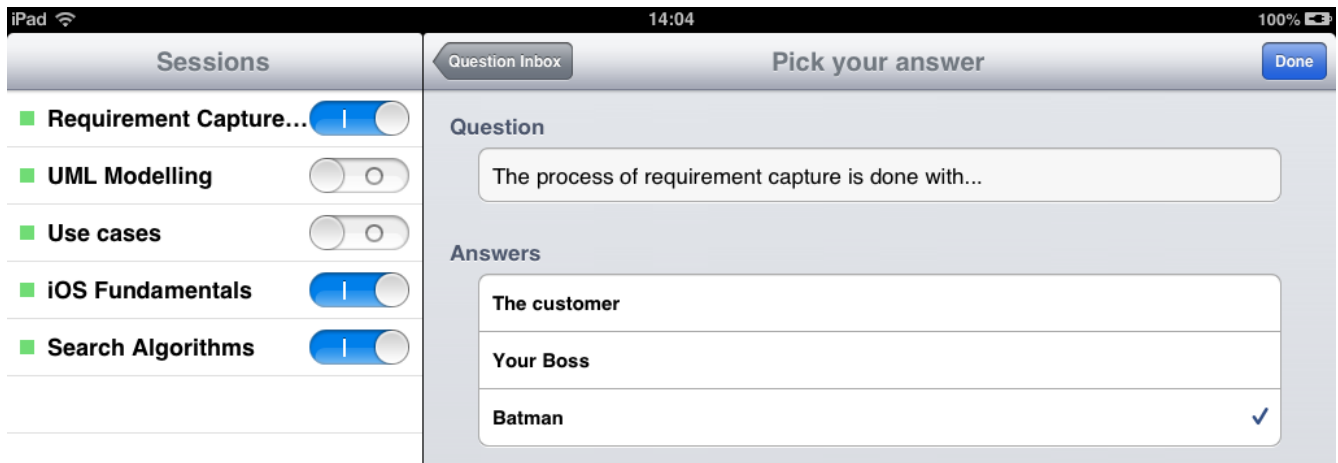


Ilustración 48: Pantalla de respuesta a preguntas

---

## **Apéndice B: Manual de mTeach Cloud Datastore**

---

## B.1 Introducción

El modelo de datos de mTeach fue inicialmente concebido para ser consumido desde la plataforma iOS y ha sido probado únicamente desde dicha plataforma, por lo tanto el empleo de dicho modelo de datos desde otras plataformas es absolutamente experimental.

El modelo de datos también está en constante evolución y en consecuencia este documento también para reflejar dichos cambios. Las aplicaciones que hagan uso del almacén de datos en la nube deberán actualizarse conforme a los cambios que aquí se establezcan.

### B.1.1 Propósito del documento

El presente documento va dirigido a los desarrolladores que desean integrar sus aplicaciones con el almacén de datos en la nube de mTeach, para ello se explicará la estructura de los datos y se dará información acerca de cómo acceder al almacén de datos para realizar lecturas y escrituras.

Se proporcionará también la información de seguridad necesaria para el acceso al API de Dropbox Datastore y al visor de datos del Datastore.

### B.1.2 Histórico de versiones

La siguiente tabla muestra el registro de revisiones del documento.

Versión	Autor	Cambios	Fecha
1.0	Roberto Estrada	Versión Inicial	01/08/13

Tabla 101: Histórico de versiones

### B.1.3 Documentación de referencia

En el presente documento no se va a detallar el proceso específico que debe seguir un desarrollador para integrar su aplicación con Dropbox Datastore. Es tarea propia del desarrollador consultar la documentación del servicio para integrarlo en su aplicación.

Dropbox Datastore actualmente sólo está disponible para las plataformas iOS, Android y JavaScript. Puede accederse a la documentación de cada plataforma desde el siguiente enlace <https://www.dropbox.com/developers/datastore>

### B.1.4 Datos de acceso al API

Los siguientes datos de acceso son comunes a todas las aplicaciones que hacen uso del almacén de datos y deben mantenerse en secreto para que únicamente las aplicaciones autorizadas puedan hacer uso del almacén.

- Dropbox APP Key: <omitido>
- Dropbox APP Secret: <omitido>

Asimismo también se emplea una cuenta Dropbox para acceder a los datos de ejemplo almacenados en la misma.

- Usuario: <omitido>
- Password: <omitido>



## B.2 Descripción del modelo de datos

El modelo de datos está compuesto por una serie de entidades relacionadas entre sí que conforman la estructura de contenidos que pueden manejar las distintas aplicaciones productoras o consumidoras de información.

Se sugiere una implementación de referencia del modelo que de hecho es la que se está empleando en la plataforma iOS a través de los framework Core Data y ParcelKit.

El modelo se compone de seis entidades como puede verse en la siguiente figura.

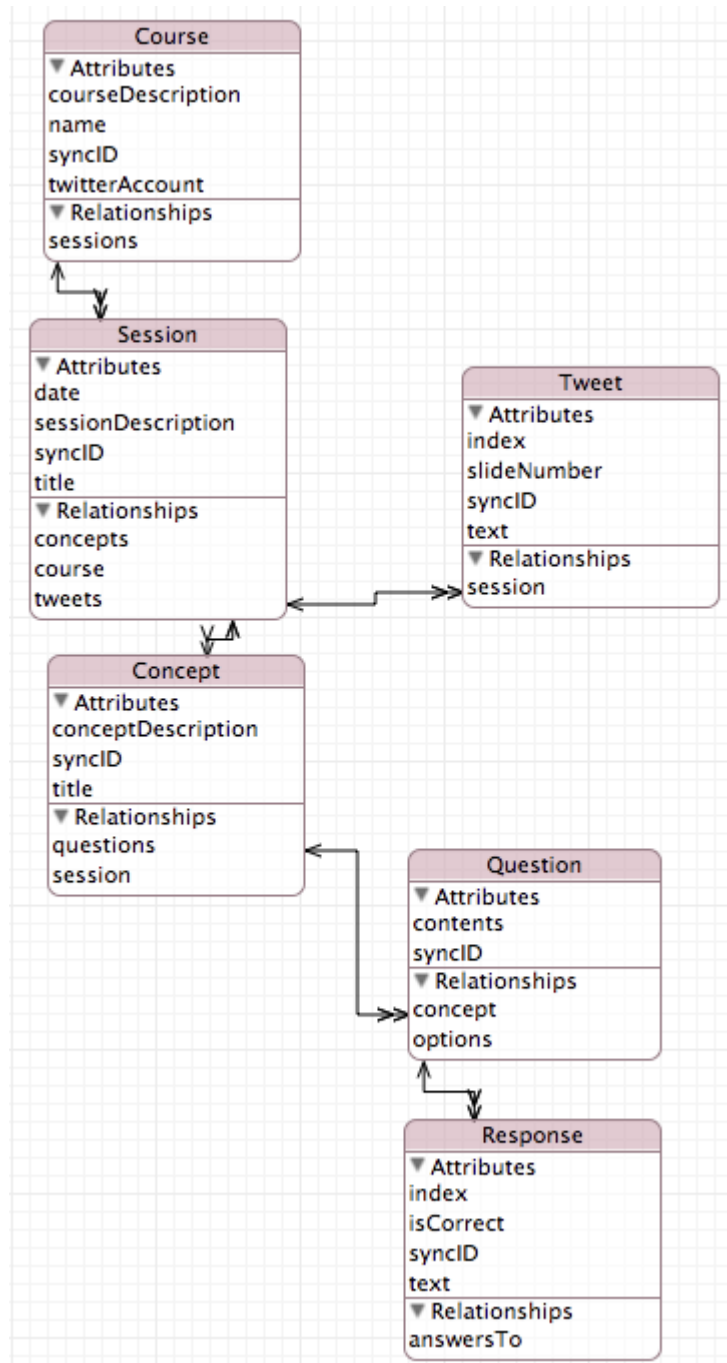


Ilustración 49: Esquema del modelo de datos empleado en iOS

## B.2.1 Claves, Integridad referencial y otras consideraciones

Las claves identificativas son automáticamente gestionadas en iOS por Core Data, por tanto queda como responsabilidad del desarrollador cómo gestionarla dentro de su aplicación.

Las relaciones entre entidades no preservan ningún tipo de orden y la regla de borrado por defecto en todos los casos es el borrado en cascada.

El atributo *syncID* es un atributo interno y autogenerado usado en iOS para relacionar los datos locales del dispositivo con los datos en Dropbox Datastore.

El atributo *index* que poseen varias entidades sirve para que el cliente pueda ordenar si así lo desea las ocurrencias a la hora de presentar los datos.

---

## B.3 Introducción a Dropbox Datastore

En este apartado se presentará Dropbox Datastore, la tecnología sobre la que se asienta mTeach Cloud Datastore.

### B.3.1 Tipos de datos soportados

Actualmente se documenta que Dropbox Datastore soporta los siguientes tipos de datos<sup>7</sup> [22].

- **String**
- **Boolean**
- **Integer** – 64 bits, signed
- **Floating point** – IEEE double
- **Date** – POSIX-like timestamp stored with millisecond precision.
- **Bytes** – Arbitrary data, which is treated as binary, such as thumbnail images and compressed data. Individual records can be up to 100KB, which limits the size of the blob. If you want to store larger files, you should use the [Sync API](#) and reference the paths to those files in your records.
- **List** – A special value that can contain other values, though not other lists. Lists are special field values. They contain an ordered list of other values, though not other lists. Lists can be manipulated via four list operations: put (i.e. replace), move, insert, and delete. These list operations allow Dropbox to handle merging changes to the structure of the list automatically.

### B.3.2 Organización del almacenamiento

El almacenamiento en Dropbox Datastore se organiza en almacenes, tablas y registros. Un almacén es un esquema de datos definido —como en cualquier base de datos relacional— por las tablas y los registros.

---

<sup>7</sup> Los tipos de datos están en constante evolución y puede consultarse información más actualizada en <https://www.dropbox.com/developers/datastore/tutorial/android#records>

## B.4 Descripción del almacén de mTeach Cloud Datastore

En este apartado se describirán las tablas que componen el almacén de datos. Todos los nombres deben tomarse literalmente haciendo distinción de mayúsculas y minúsculas.

### B.4.1 Tabla courses

Campo	Descripción	Tipo de datos
<b>id</b>	Clave primaria (autogenerada por Dropbox)	String (Solo lectura)
<b>twitterAccount</b>	Cuenta twitter del curso	String
<b>courseDescription</b>	Descripción del curso	String
<b>name</b>	Nombre del curso	String
<b>sessions</b>	Sesiones asociadas a un curso	List (Lista de id)

Tabla 102: Tabla courses

### B.4.2 Tabla sessions

Campo	Descripción	Tipo de datos
<b>id</b>	Clave primaria (autogenerada por Dropbox)	String (Solo lectura)
<b>sessionDescription</b>	Descripción de la sesión	String
<b>date</b>	Fecha de la sesión	Date
<b>title</b>	Título (nombre) de la sesión	String
<b>course</b>	Id del curso al que pertenece (Clave ajena)	String
<b>concepts</b>	Conceptos asociados a una sesión	List (Lista de id)
<b>tweets</b>	Tweets asociados a una sesión	List (Lista de id)

Tabla 103: Tabla sessions

### B.4.3 Tabla concepts

Campo	Descripción	Tipo de datos
<b>id</b>	Clave primaria (autogenerada por Dropbox)	String (Solo lectura)
<b>conceptDescription</b>	Descripción del curso	String
<b>session</b>	Id de la sesión a la que pertenece (Clave ajena)	String
<b>questions</b>	Preguntas asociadas a un concepto	List (Lista de id)

Tabla 104: Tabla concepts

### B.4.4 Tabla tweets

Campo	Descripción	Tipo de datos
<b>id</b>	Clave primaria (autogenerada por Dropbox)	String (Solo lectura)
<b>text</b>	Contenido del tweet (No deben almacenarse más de 140 caracteres)	String
<b>slideNumber</b>	Número de transparencia a la que está ligado el tweet (opcional)	Integer
<b>session</b>	Id de la sesión a la que pertenece (Clave ajena)	String
<b>index</b>	Número para especificar orden de presentación en el cliente (opcional)	Integer

Tabla 105: Tabla tweets

### B.4.5 Tabla questions

Campo	Descripción	Tipo de datos
<b>id</b>	Clave primaria (autogenerada por Dropbox)	String (Solo lectura)
<b>contents</b>	Enunciado de la pregunta	String
<b>concept</b>	Id del concepto al que pertenece (Clave ajena)	String
<b>options</b>	Respuestas asociadas a una sesión	List (Lista de id)

Tabla 106: Tabla questions

### B.4.6 Tabla responses

Campo	Descripción	Tipo de datos
<b>id</b>	Clave primaria (autogenerada por Dropbox)	String (Solo lectura)
<b>text</b>	Texto de la opción de respuesta	String
<b>answersTo</b>	Id de la pregunta a la que responde (Clave ajena)	String
<b>isCorrect</b>	Indica si la ocurrencia es una opción de respuesta correcta o no.	Integer (0=falso, 1=verdadero).
<b>index</b>	Número para especificar orden de presentación en el cliente (opcional)	Integer

Tabla 107: Tabla responses